Fundamenta Informaticae XX (2007) 1–44
IOS Press

Causal Semantics of Algebraic Petri Nets distinguishing Concurrency and Synchronicity

Gabriel Juhás
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology Bratislava
gabriel.juhas@stuba.sk

Robert Lorenz
Department of Applied Computer Science
Catholic University of Eichstätt-Ingolstadt
robert.lorenz@ku-eichstaett.de

Sebastian Mauser
Department of Applied Computer Science
Catholic University of Eichstätt-Ingolstadt
sebastian.mauser@ku-eichstaett.de

Abstract. In this paper, we show how to obtain causal semantics distinguishing "earlier than" and "not later than" causality between events from algebraic semantics of Petri nets.

Janicki and Koutny introduced so called stratified order structures (so-structures) to describe such causal semantics. To obtain algebraic semantics, we redefine our own algebraic approach generating rewrite terms via partial operations of synchronous composition, concurrent composition and sequential composition. These terms are used to produce so-structures which define causal behavior consistent with the (operational) step semantics. For concrete Petri net classes with causal semantics derived from processes minimal so-structures obtained from rewrite terms coincide with minimal so-structures given by processes. This is demonstrated for elementary nets with inhibitor arcs.

Keywords: theory of concurrency, algebraic Petri nets, causal semantics, process terms, inhibitor arcs, synchronicity

Address for correspondence: Sebastian Mauser, Department of Applied Computer Science, Catholic University of Eichstätt-Ingolstadt, sebastian.mauser@ku-eichstaett.de

1. Introduction

Since the basic developments of Petri nets more and more different *Petri net classes* for various applications have been proposed. Causal semantics of such special Petri net classes are often constructed in a complicated ad-hoc way, defining process nets which generate causal structures (see e.g. [20, 10, 15, 16]).

Naturally there are also several approaches to unify the different classes in order to be able to define non-sequential semantics in a systematic way using algebraic descriptions [25, 3, 1, 8, 23, 21, 22] (see [24] for an overview). Most of these approaches are based on the paper [19], where non-sequential runs of nets are described by equivalence classes of rewrite process terms. These process terms are generated from elementary terms (transitions and markings) by concurrent and sequential composition. Unfortunately, none of these works provides a method how to obtain causal semantics from the algebraic semantics.

This paper extends the *unifying approach* of *algebraic Petri nets* as proposed in Part II of [11]. With the approach from [11] *non-sequential semantics* can be derived on an abstract level for Petri nets with restricted occurrence rule (encoded by partiality of concurrent composition) like place/transition nets (p/t-nets) with capacities, elementary nets with mixed context equipped with the a-posteriori semantics, etc. In addition to other works, and in particular to [8], in [11] it is shown how to obtain causal semantics based on an "earlier than" causality between events (formally given as labelled partial orders (LPOs)) from process terms. It is proven in [11] for many concrete net classes that the minimal LPOs obtained from process terms coincide with minimal LPOs given by standard process semantics.

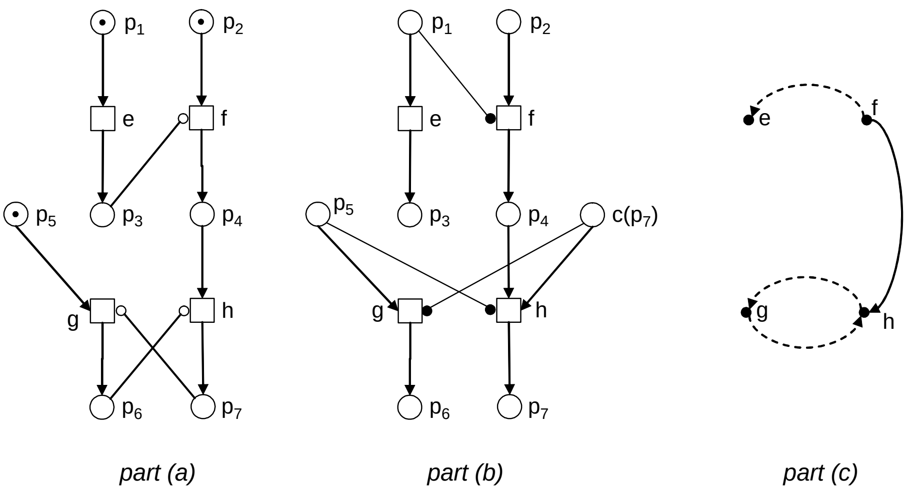

Figure 1. An elementary net with inhibitor arcs (p_3, f), (p_6, h) and (p_7, g) (part (a)), a process of the net (part (b)) and the associated run (part (c)).

As explained in [10], "earlier than" causality expressed by LPOs is not enough to describe causal semantics for some Petri net classes, as for example the a-priori semantics of elementary nets with inhibitor arcs.[1] In Figure 1 (which serves as our running example) this phenomenon is depicted: In the a-priori semantics the testing for absence of tokens (through inhibitor arcs) precedes the execution of a

[1]Note that there are also other semantics for elementary nets with inhibitor arcs [5, 26].

transition. Thus f cannot occur later than e, because after the occurrence of e the place p_3 is marked and consequently the occurrence of f is prohibited by the inhibitor arc (p_3, f). Therefore e and f cannot occur concurrently or sequentially in the order $e \to f$. But they still can occur synchronously (because of the occurrence rule "testing before execution") or sequentially in order $f \to e$. This is exactly the behavior described by "f not later than e" (see section 2 for details on the occurrence rule). After the respective firing of f and e we reach the marking $\{p_3, p_4, p_5\}$. Now with the same arguments as above the transitions g and h can occur synchronously but not sequentially in any order. This relationship can be expressed by a symmetric "not later than" relation between the respective events (none may occur later than the other). The described causal behavior (between events) of the net is illustrated in part (c) of Figure 1. The solid arcs represent a (common) "earlier than" relation, i.e. the events can only occur in the expressed order but not synchronously or inversely. Dashed arcs depict the "not later than" relation explained above. Partial orders only model the "earlier than" relation but they do not permit to describe relationships, where synchronous occurrence is possible but concurrency is not. Examples are the relationships between e and f as well as g and h in Figure 1. The net in part (b) of Figure 1 depicts a process corresponding to the run in part (c) (details on processes and runs are explained in section 2). Altogether there exist net classes, including inhibitor nets admired by practitioners, where synchronous and concurrent behaviour have to be distinguished.[2] In [10] causal semantics based on stratified order structures (so-structures, see section 2) consisting of a combination of an "earlier than" and a "not later than" relation between events were proposed to cover such cases. The run in Figure 1 illustrates such an so-structure describing the causalities explained in this paragraph.

In order to describe such situations at the algebraic level, in [13] we extended the algebraic Petri nets from [8] by a synchronous composition operation which allows distinguishing between concurrent and synchronous occurrences of events. In such an algebraic approach, a transition t is understood to be an elementary rewrite term allowing to replace the (initial) marking $pre(t)$ by the (final) marking $post(t)$. Moreover, any marking m is understood to be an elementary term, rewriting m by m itself. A single occurrence of a transition t leading from a marking m to a marking m' can be understood as a concurrent composition of the elementary term t and the elementary term corresponding to the marking x, satisfying $m = x + pre(t)$ and $m' = x + post(t)$, where $+$ denotes a suitable operation on markings (see Figure 2). The non-sequential behaviour of a net is given by a set of process terms constructed from elementary terms as follows: Firstly transitions can be synchronously composed to *synchronous step terms* using an operator $\oplus$ for synchronous composition (in particular every transition itself is a synchronous step term). Secondly markings and synchronous step terms can be sequentially and concurrently composed to (general) *process terms* using operators for sequential and for concurrent composition, denoted by ; and $\|$, respectively.

As described in the running example, transitions t and t' cannot necessarily occur synchronously resp. concurrently at the marking $pre(t) + pre(t')$. Such restrictions of the occurrence rule will be encoded by a restriction of synchronous and concurrent composition. That means if the marking $pre(t) + pre(t')$ does not enable two transitions t and t' synchronously (concurrently), then t and t' are not allowed to be composed by $\oplus$ ($\|$). To describe such a restriction, we use an abstract set I of information elements, together with two symmetric independence relations on I for synchronous and concurrent composition. Every marking x as well as every transition t has an attached information element. Several transitions

[2] Further examples of such net classes are nets with read arcs (a-priori semantics), nets with capacities ("first consume then produce" semantics), nets with priorities, nets with reset arcs, nets with signal arcs, etc.

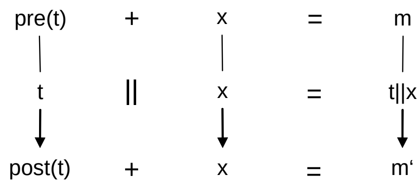

Figure 2. Firing of a transition t from a marking m to a marking m' and its interpretation as a concurrent rewriting of the transition t and the marking x.

can be composed synchronously (to synchronous step terms) if and only if their respective information elements are (synchronously) independent. For synchronously independent information elements we define an operation for the synchronous composition of information elements. This operation has the meaning that the information of the composed synchronous step term is the composition of the information elements of its components. A marking x and a synchronous step term s can be composed concurrently if and only if their respective information elements are (concurrently) independent. We also define an operation for the concurrent composition of information elements with the intended meaning that the information of the composed term is the (concurrent) composition of the information elements of its components. Since the operations of synchronous and concurrent composition between elementary terms and information elements are defined only partially, i.e. partial algebra is employed, such nets are also called Petri nets over partial algebra [8].

As elementary terms, each process term has an associated initial marking, final marking and an information set consisting of all information elements of elementary terms from which it is generated. Initial and final markings are necessary for sequential composition: Two process terms can be composed sequentially only if the final marking of the first process term coincides with the initial marking of the second one. Then the initial marking of the resulting process term is the initial marking of the first term and the final marking of the resulting term is the final marking of the second term. The set of information associated to the resulting process term is given by the union of the sets of information associated to the two composed terms. Concurrent composition of two process terms is defined only if each element of the associated information set of the first process term is independent from each element of the information set of the second term. Then the initial and final marking of the resulting term are given by the sum $(+)$ of the initial and final markings of the two terms. The set of information of the resulting process term contains the concurrent composition of each element of the information set of the first term with each element of the information set of the second. Synchronous composition is only defined for transitions but not for arbitrary process terms.

Altogether, an algebraic Petri net consists of a set of transitions and a set of markings (equipped with an operation for the addition of markings). Each transition is assigned an initial and a final marking and each marking and each transition is assigned an information element from a partial algebra of information. Its behaviour is given by process terms. Process terms are built inductively by synchronous composition of transitions to synchronous step terms, and sequential and concurrent composition of synchronous step terms and markings. All composition operations are partial. In this approach, the cardinality of the sets of information associated to concurrently composed process terms growths exponentially. For deriving a more compact information one can use any equivalence $\cong \in 2^I \times 2^I$ that is a congruence with respect to the operations concurrent composition and union for sequential composition (the synchronous composition operation is not considered since it is not defined for arbitrary process terms). If this congruence preserves the independence relation, i.e. $A \cong B$ and A is independent from C (that means each element of A is independent from each element of C) implies that B is independent from C, then the congruence is called a *closed congruence*. Equivalence classes of the greatest closed congruence represent the minimal information assigned to process terms necessary for concurrent composition. Thus, instead of sets of information we associate to process terms equivalence classes w.r.t. the greatest closed congruence.

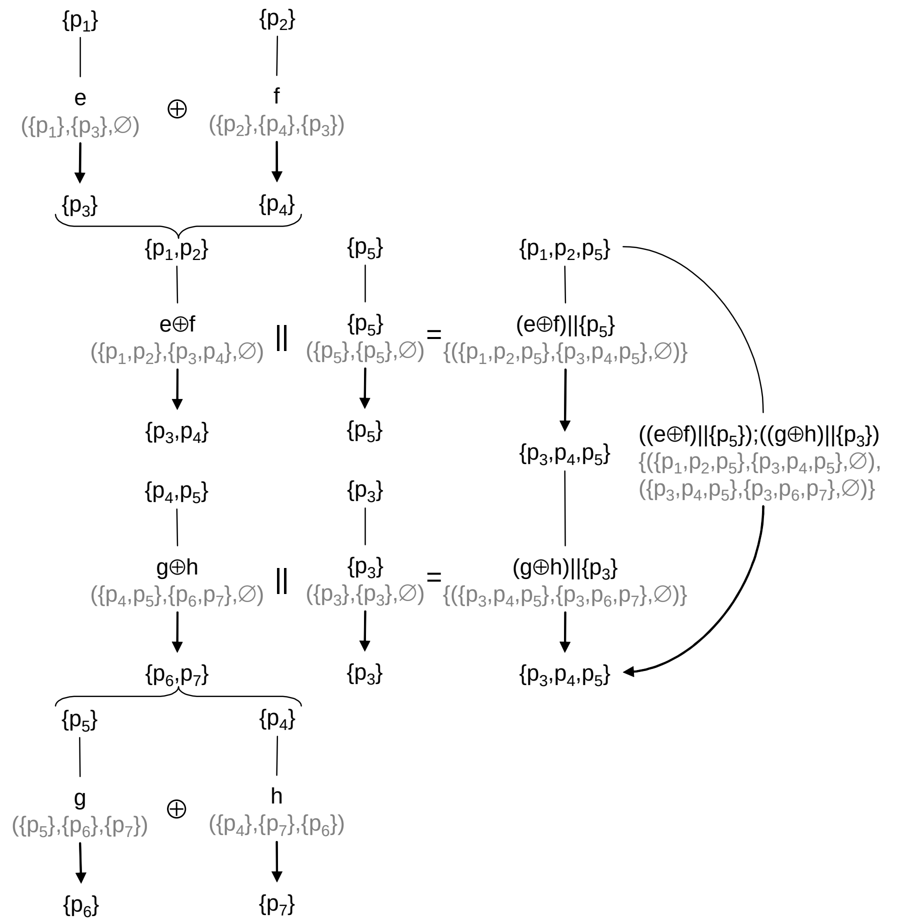

Figure 3. Deriving an exemplary process term of the net from Figure 1 (part (a)). In the middle of an arc there are drawn the respective sub-process terms with associated information (information elements for elementary process terms and information sets for non-elementary process terms) in the line below in grey colour. At the beginning of an arrow we illustrated pre and at the arrowhead $post$ is depicted.

Figure 3 shows an example for the construction of a process term to describe a run of the net shown in Figure 1, part (a). In the initial marking $\{p_1, p_2, p_5\}$ the transitions e and f can occur synchronously. This is described by the process term $(e \oplus f) \parallel \{p_5\}$. After the occurrence of e and f the marking $\{p_3, p_4, p_5\}$ is reached. In this marking, the transitions g and h can occur synchronously yielding the process term $(g \oplus h) \parallel \{p_3\}$. The resulting term is $((e \oplus f) \parallel \{p_5\}); ((g \oplus h) \parallel \{p_3\})$. For the construction of such a process term one has to verify that the mentioned synchronous and concurrent compositions of subterms are defined. This is done by assigning information elements from an appropriate partial algebra of information to markings and transitions as described above (see Section 6 and Example 3.1 for the formal definition of such a partial algebra of information for elementary nets with inhibitor arcs).

Using the described algebraic approach, a great variety of additional concrete net classes can be covered compared to [11]. Unfortunately, the paper [13] does not provide a general method for constructing so-structure based causal semantics from the algebraic semantics. Therefore in [13] a correspondence of the algebraic semantics to non-sequential a-priori process semantics of elementary nets with inhibitor arcs was proven in a complicated ad hoc way not comparing causal semantics.

As the main result of this paper we fill this gap. Namely, we show how to obtain a causal semantics based on so-structures from process terms and, as an example, derive their correspondence to causal semantics produced from processes for elementary nets with inhibitor arcs equipped with the a-priori semantics.

Causal semantics can be obtained from process terms as follows: First each process term α defines an so-structure (whose events are labelled by transition occurrences) in an obvious way:

- an event e_1 occurs *earlier than* another event e_2 if the process term α contains a subterm $\alpha_1; \alpha_2$ such that e_1 occurs in α_1 and e_2 occurs in α_2.
- events of one synchronous step term are in symmetric *not later than* relation.

For example, the process term $((e \oplus f) \parallel \{p_5\}); ((g \oplus h) \parallel \{p_3\})$ generates the left most so-structure shown in Figure 4.

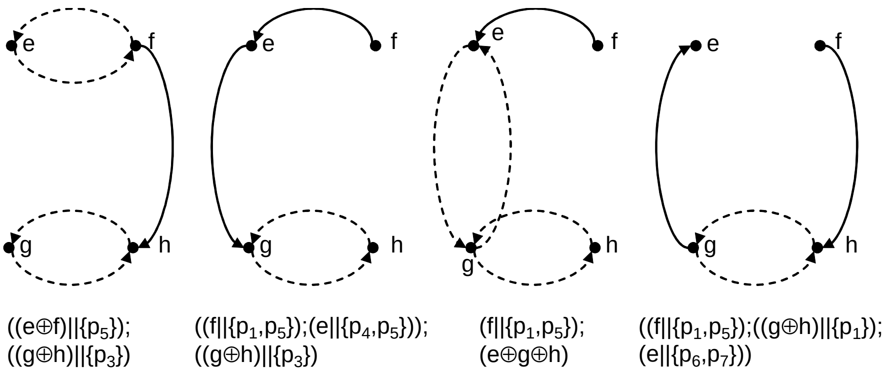

Figure 4. So-structures generated from process terms describing possible behaviour of the net from Figure 1 (part (a)).

Unfortunately not all reasonable so-structures can be generated in this way. For example, consider the so-structure shown in Figure 1, part (c). It is easy to show by induction on the structure of process terms that this so-structure cannot be generated by any process term. However, this so-structure can be constructed from the so-structures shown in Figure 4 (by intersection of the respective "earlier than" and "not later than" relations) which are all generated by process terms.

Formally, process terms are used to produce so called *enabled* so-structures defining causal semantics from algebraic semantics of Petri nets. These causal semantics are consistent with the algebraic step semantics in the sense that an so-structure is enabled if and only if every of its step sequentializations is generated by an appropriate process term (Figure 4 shows all step sequentializations of the so-structure in Figure 1, part (c)). Of course, this causal semantics should coincide with causal semantics of concrete Petri net classes derived from processes (for such Petri net classes which have defined process semantics). That means, the minimal enabled so-structures obtained from process terms should be exactly minimal so-structures given by processes. This property must be shown extra for each concrete Petri net class. In order to prepare the possibility of a systematic proof at such a concrete level, we derive results at the abstract algebraic level relating process terms and enabled so-structures in more detail. Firstly it is shown that so-structures generated by process terms are enabled. Secondly, we will define an equivalence of process terms which are intended to represent the same commutative process (in commutative processes one abstracts from the individuality of tokens, for details and examples see [11] and [2])[3]. Then it can be proven that two process terms, whose generated so-structures extend (add causality to) the same enabled so-structure, are equivalent.

Abstracting from technical details the framework for non-sequential semantics described in this paper is visualized in Figure 5. Analogous relations were developed in [11] for the partial order case without a synchronous composition operator. Here we elaborate the relations in this more general situation, where our main focus is the causal level. Note that the right side (the filled arcs) of the graph - deriving runs using the algebraic framework - can be schematically applied to nearly any net class with restricted occurrence rule. The left side (non-filled arcs) of the framework is obviously strongly connected to the concrete process definitions which are often, as already mentioned, non uniform ad-hoc definitions. Consequently to demonstrate the respective relations we have to discuss a concrete net class implementation as it is shown in section 6. But our approach is not dependent on such process semantics and we are able to define runs for a wide variety of net classes in a uniform way (filled arcs): The framework directly provides non-sequential algebraic semantics (process terms) and based on these causal semantics (runs in the form of so-structures).

Formally, the algebraic semantics of a concrete Petri net is given as follows: We say that an algebraic Petri net *corresponds* to a concrete Petri net, if both nets have the same (algebraic respectively operational) step semantics. Given a concrete Petri net, first a corresponding algebraic net has to be determined. For this we have to fix an appropriate partial algebra of information and assign information elements to markings and transitions. Then process terms can be generated from which (minimal) enabled so-structures can be deduced in a systematic way. The resulting semantics are algebraically established, are consistent with operational step semantics, and exactly match acknowledged classical semantics in all elaborated examples. The advantage over classical non-sequential semantics is the uniform, systematic and mathematically founded definition.

The paper is structured as follows: First we generalize our own algebraic approach from [13] generating process terms via the partial operations of synchronous composition, concurrent composition and sequential composition (section 3). These terms are used to produce so called *enabled* so-structures defining causal semantics of algebraic nets. We derive several general results establishing the detailed relationship between process terms and enabled so-structures at this abstract algebraic level (section 4). Given a Petri net of a concrete Petri net class, we define the corresponding algebraic net in section 5. This leads to a systematic and uniform way to construct causal semantics of Petri nets. This causal semantics

[3]Note that for elementary nets (with inhibitor arcs) we do not have to distinguish between collective and individual token semantics (see [11]) and thus commutative processes and classical processes coincide.

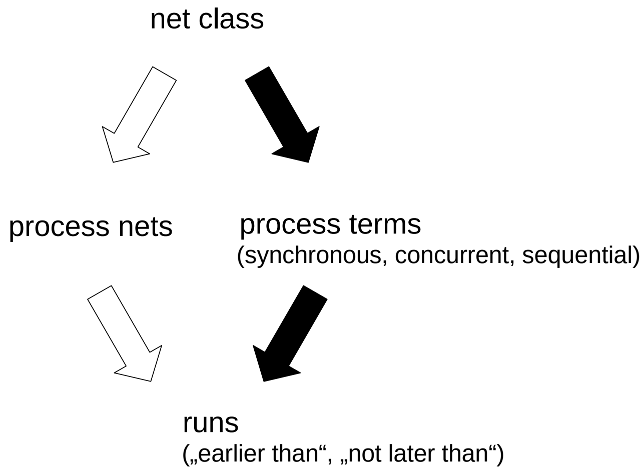

Figure 5. The semantical framework of this paper which sketches the relations between concrete Petri net classes (possibly distinguishing synchronous and concurrent behaviour), classical process definitions based on occurrence nets, algebraic process terms with the partial operations of synchronous, concurrent and sequential composition and in the end runs expressed as labelled so-structures.

coincides with classical causal semantics derived from process semantics. Exemplarily we will show this result in a systematic way (which is based on the general results developed in section 4 and can be adapted to further net classes) for elementary nets with inhibitor arcs (section 6).[4] This generalizes the main result of [13].

Altogether this work is a self-contained, elaborated and consolidated version of [12]. For better readability we only sketched the technically laborious proofs in the main text of the paper but we included detailed proofs in the Appendix. If we do not explicitly mention that a result was already proven in another publication, then the result is firstly shown in this paper. The same holds for the conceptual definitions.

2. Preliminaries

In this section we recall the basic definitions of *stratified order structures*, *elementary nets with inhibitor arcs (equipped with the a-priori semantics)* and *partial algebras*.

Given a set X we will denote the set of all subsets of X by 2^X, the set of all multisets over X by $\mathbb{N}^X$, the identity relation over X by id_X, the reflexive transitive closure of a binary relation R over X by R^* and the composition of two binary relations R, R' over X by $R \circ R'$.

We start with some basic notions to prepare the ground for the definition of *stratified order structures* (*so-structures*). A *directed graph* is a pair $(V, \rightarrow)$, where V is a finite *set of nodes* and $\rightarrow \subseteq V \times V$ is a binary relation over V called the *set of arcs*. As usual, given a binary relation $\rightarrow$, we write $a \rightarrow b$ to

[4] This net class has the advantage that it is already completely analyzed in the concept of ad-hoc process definitions (in contrast to p/t-nets with weighted or unweighted inhibitor arcs [15, 16]). Therefore we are able to check the consistency of the ad-hoc concept to our general algebraic concept. Note that the existing process semantics for general p/t-nets with weighted inhibitor arcs [16] do not produce minimal causal behaviour.

denote $(a,b) \in \rightarrow$. Two nodes $a, b \in V$ are called *independent* w.r.t. the binary relation $\rightarrow$ if $a \not\rightarrow b$ and $b \not\rightarrow a$. We denote the set of all pairs of nodes independent w.r.t. $\rightarrow$ by $\mathrm{co}_{\rightarrow} \subseteq V \times V$. A *partial order* is a directed graph $\mathrm{po} = (V, <)$, where $<$ is an irreflexive and transitive binary relation on V. If $\mathrm{co}_{<} = id_V$ then $(V, <)$ is called *total*. A co-set of a partial order $(V, <)$ is a set C satisfying $C \times C \subseteq \mathrm{co}_{<}$. Given two partial orders $\mathrm{po}_1 = (V, <_1)$ and $\mathrm{po}_2 = (V, <_2)$, we say that po_2 is a *sequentialization* (or *extension*) of po_1 if $<_1 \subseteq <_2$.

So-structures [10] are, loosely speaking, combinations of two binary relations on a set of events where one is a partial order representing an "earlier than" relation and the other represents a "not later than" relation. Thus so-structures describe finer causalities than partial orders. Formally, so-structures are relational structures satisfying certain properties. A *relational structure* (rel-structure) is a triple $\mathcal{S} = (X, \prec, \sqsubset)$, where X is a finite set (of *events*), and $\prec \subseteq X \times X$ and $\sqsubset \subseteq X \times X$ are binary relations on X. A rel-structure $\mathcal{S}' = (X, \prec', \sqsubset')$ is said to be an *extension* of another rel-structure $\mathcal{S} = (X, \prec, \sqsubset)$, written $\mathcal{S} \subseteq \mathcal{S}'$, if $\prec \subseteq \prec'$ and $\sqsubset \subseteq \sqsubset'$.

Definition 2.1. (Stratified order structure)
A rel-structure $\mathcal{S} = (X, \prec, \sqsubset)$ is called *stratified order structure* (so-structure) if the following conditions are satisfied for all $x, y, z \in X$:
(C1) $x \not\sqsubset x$
(C2) $x \prec y \Longrightarrow x \sqsubset y$
(C3) $x \sqsubset y \sqsubset z \wedge x \neq z \Longrightarrow x \sqsubset z$
(C4) $x \sqsubset y \prec z \vee x \prec y \sqsubset z \Longrightarrow x \prec z$ ○

In figures $\prec$ is graphically expressed by solid arcs and $\sqsubset$ by dashed arcs. According to (C2) a dashed arc is omitted if there is already a solid arc. Moreover, we omit arcs which can be deduced by (C3) and (C4). It is shown in [10] that $(X, \prec)$ is a partial order ($(X, \sqsubset)$ is a strict preorder). Therefore so-structures are a generalization of partial orders. So-structures turned out to be adequate to model the causal relations between events of complex systems exhibiting sequential, concurrent and synchronous behavior. In this context $\prec$ represents the ordinary "earlier than" relation (as in partial order based systems) while $\sqsubset$ models a "not later than" relation (examples are depicted in Figure 1, part (c), and Figure 4). According to [10] for nodes $x, y \in X$ there is an extension $\mathcal{S}' = (X, \prec', \sqsubset')$ of $\mathcal{S}$ with $x \prec' y$ if and only if $y \not\sqsubset x$ and $x \neq y$. Moreover, for all $x, y \in X$, there holds $x \prec y \Longrightarrow y \not\sqsubset x$. These properties justify the described interpretation of $\prec$ and $\sqsubset$.

Similar to the notion of the transitive closure of a binary relation the *$\diamondsuit$-closure* $\mathcal{S}^{\diamondsuit}$ [10] of a rel-structure $\mathcal{S} = (X, \prec, \sqsubset)$ is defined by

$$\begin{aligned} \mathcal{S}^{\diamondsuit} &= (X, \prec_{\mathcal{S}^{\diamondsuit}}, \sqsubset_{\mathcal{S}^{\diamondsuit}}) \\ &= (X, (\prec \cup \sqsubset)^{*} \circ \prec \circ (\prec \cup \sqsubset)^{*}, (\prec \cup \sqsubset)^{*} \setminus id_X). \end{aligned}$$

A rel-structure $\mathcal{S}$ is called *$\diamondsuit$-acyclic* if $\prec_{\mathcal{S}^{\diamondsuit}}$ is irreflexive. The *$\diamondsuit$-closure* $\mathcal{S}^{\diamondsuit}$ of a rel-structure $\mathcal{S}$ is an so-structure if and only if $\mathcal{S}$ is $\diamondsuit$-acyclic (for this and further results on the $\diamondsuit$-closure see [10]).

Finally, we introduce two subclasses of so-structures which turn out to be associated to (specific subclasses of) process terms of algebraic Petri nets.

Definition 2.2. Let $\mathcal{S} = (X, \prec, \sqsubset)$ be an so-structure, then $\mathcal{S}$ is called *synchronous closed* if $\mathrm{co}_{\prec} = \mathrm{co}_{\sqsubset} \cup (\sqsubset \setminus \prec)$ and $\mathcal{S}$ is called *total linear* if $\mathrm{co}_{\prec} = (\sqsubset \setminus \prec) \cup id_X$. ○

In other words, a synchronous closed so-structure can be regarded as an so-structure that does not include asymmetric $\sqsubset$-relations (for instance the so-structure in Figure 1, part (c), is not synchronous closed, because $f \sqsubset e$ but $e \not\sqsubset f$). Total linear so-structures are maximally sequentialized in the sense that no further $\prec$- or $\sqsubset$- relations can be added maintaining the requirements of so-structures according to Definition 2.1 (for examples see Figure 4).

The set of all total linear extensions (or *linearizations*) of an so-structure $\mathcal{S}$ is denoted by $strat_{sos}(\mathcal{S})$. Now we will summarize some results about these two classes of so-structures. The following result proven in [15][5] shows that every so-structure can be reconstructed from its linearizations:

Proposition 2.1. Let $\mathcal{S}$ be an so-structure. Then

$$\mathcal{S} = (X, \bigcap_{(X,\prec,\sqsubset)\in strat_{sos}(\mathcal{S})} \prec, \bigcap_{(X,\prec,\sqsubset)\in strat_{sos}(\mathcal{S})} \sqsubset)$$

This means any so-structure equals the intersection of its linearizations (as an example, the so-structure in Figure 1, part (c), equals the intersection of those in Figure 4).

Each total linear so-structure is synchronous closed because according to (C2) $\mathrm{co}_\prec = (\sqsubset \setminus \prec) \cup id_X$ implies $\mathrm{co}_\sqsubset = id_X$. Using the results from [10] about augmenting so-structures one can conclude that every so-structure is extendable to a total linear so-structure.

The crucial property of synchronous closed so-structures is the fact that every synchronous closed so-structure can be *embedded* into a partial order. To show this we need the following lemma which is fundamental to transforming special equivalence classes of nodes of a synchronous closed so-structure into the nodes of a partial order:

Lemma 2.1. Let $\mathcal{S} = (X, \prec, \sqsubset)$ be an so-structure. Then $\mathcal{S}$ is synchronous closed if and only if $(\sqsubset \setminus \prec) \cup id_X$ is an equivalence relation.

Proof:
"$\Longrightarrow$": The reflexivity of $(\sqsubset \setminus \prec) \cup id_X$ is obvious. From $\mathrm{co}_\prec = \mathrm{co}_\sqsubset \cup (\sqsubset \setminus \prec)$ we deduce that $(\sqsubset \setminus \prec)$ is symmetric, since $\mathrm{co}_\prec$ and $\mathrm{co}_\sqsubset$ are symmetric and $\mathrm{co}_\sqsubset \cap (\sqsubset \setminus \prec) = \emptyset$. Consequently, $(\sqsubset \setminus \prec) \cup id_X$ is symmetric. Denote $\sim = (\sqsubset \setminus \prec) \cup id_X$. According to (C3), $\sqsubset \cup id_X$ is transitive. The transitivity of $\sim$ can be proven as follows: Assuming $x \sim y \sim z$ and $x \not\sim z$ then implies $x \prec z$. This contradicts the symmetry of $\sim$, since $x \prec z \Longrightarrow z \not\sqsubset x$.

"$\Longleftarrow$": Since $(\sqsubset \setminus \prec) \cup id_X$ is symmetric, also $(\sqsubset \setminus \prec)$ is symmetric. Therefore, $x \ \mathrm{co}_\prec y$ implies that either $x \ \mathrm{co}_\sqsubset \ y$ or $x \ (\sqsubset \setminus \prec) \ y$ and $y \ (\sqsubset \setminus \prec) \ x$. On the other hand, we get $\mathrm{co}_\sqsubset \subseteq \mathrm{co}_\prec$ directly from (C2) and $(\sqsubset \setminus \prec) \subseteq \mathrm{co}_\prec$ since $(\sqsubset \setminus \prec)$ is symmetric. □

For a synchronous closed so-structure $\mathcal{S} = (X, \prec, \sqsubset)$ we denote

- $\sim_\mathcal{S} = (\sqsubset \setminus \prec) \cup id_X$ (which according to Lemma 2.1 defines an equivalence relation),
- $[x]_\mathcal{S} = \{y \in X \mid x \sim_\mathcal{S} y\}$, and
- $X|_\mathcal{S} = \{[x]_\mathcal{S} \mid x \in X\}$.

[5]formulated in other notations using the notion $strat(\mathcal{S})$

The elements of $X|_{\mathcal{S}}$ are called *synchronous classes of* $\mathcal{S}$. The partial order $\prec$ carries over to $X|_{\mathcal{S}}$ as follows: For $[x]_{\mathcal{S}}, [y]_{\mathcal{S}} \in X|_{\mathcal{S}}$ define

$$[x]_{\mathcal{S}} <_{\mathcal{S}} [y]_{\mathcal{S}} \Longleftrightarrow x \prec y.$$

By (C4) this is well-defined. Then $po_{\mathcal{S}} = (X|_{\mathcal{S}}, <_{\mathcal{S}})$ defines a partial order, because $(X, \prec)$ is a partial order. $po_{\mathcal{S}}$ is called *associated to* $\mathcal{S}$. The partial orders associated to the total linear so-structures in Figure 4 are the total orders expressed by the following sequences (from left to right): $\{e, f\} \to \{g, h\}$, $\{f\} \to \{e\} \to \{g, h\}$, $\{f\} \to \{e, g, h\}$ and $\{f\} \to \{g, h\} \to \{e\}$ (this also illustrates the second statement in the subsequent Lemma 2.2). We have the following results for associated partial orders:

Lemma 2.2. Let $\mathcal{S}, \mathcal{S}'$ be synchronous closed so-structures satisfying $\mathcal{S} \subseteq \mathcal{S}'$. Then $k' \in X|_{\mathcal{S}'}$ has the form $k' = k_1 \cup \ldots \cup k_n$ with $k_1, \ldots, k_n \in X|_{\mathcal{S}}$ for some $n \in \mathbb{N}$.

Moreover, a synchronous closed so-structure $\mathcal{S}$ is total linear if and only if its associated partial order $po_{\mathcal{S}}$ is total.

Proof:
From the above definitions we directly deduce $\sim_{\mathcal{S}} \subseteq \sim_{\mathcal{S}'}$. This implies the first statement of the Lemma.

For the second statement we have to discuss two directions:

Let $\mathcal{S}$ be total linear, i.e. $\mathrm{co}_{\prec} = (\sqsubset \setminus \prec) \cup id_X = \sim_{\mathcal{S}}$. This implies $x \,\mathrm{co}_{\prec} y \Longleftrightarrow [x]_{\mathcal{S}} = [y]_{\mathcal{S}}$ $(x, y \in X)$. This gives $\mathrm{co}_{<} = id_{X|_{\mathcal{S}}}$ because $x \,\mathrm{co}_{\prec} y \Longleftrightarrow [x]_{\mathcal{S}} \,\mathrm{co}_{<} [y]_{\mathcal{S}}$ by definition.

Let on the other hand $po_{\mathcal{S}}$ be total, i.e. $\mathrm{co}_{<} = id_{X|_{\mathcal{S}}}$. From $x \,\mathrm{co}_{\prec} y \Longleftrightarrow [x]_{\mathcal{S}} \,\mathrm{co}_{<} [y]_{\mathcal{S}}$ we deduce $x \,\mathrm{co}_{\prec} y \Longleftrightarrow [x]_{\mathcal{S}} = [y]_{\mathcal{S}}$, i.e. $\mathrm{co}_{\prec} = \sim_{\mathcal{S}} = (\sqsubset \setminus \prec) \cup id_X$. □

We will often use *labelled so-structures* in the following. These are so-structures $\mathcal{S} = (X, \prec, \sqsubset)$ together with a *set of labels* M and a *labelling function* $l : X \to M$. We use the above notations defined for so-structures also for labelled so-structures. Moreover for labelled so-structures we define: Two labelled so-structures $(V_1, \prec_1, \sqsubset_1, l_1), (V_2, \prec_2, \sqsubset_2, l_2)$ are isomorphic if and only if there is a bijection $\gamma : V_1 \to V_2$ preserving the order relations and the labelling function, i.e. $\forall v_1, v_2 \in V_1 : v_1 \prec_1 v_2 \Longleftrightarrow \gamma(v_1) \prec_2 \gamma(v_2) \wedge v_1 \sqsubset_1 v_2 \Longleftrightarrow \gamma(v_1) \sqsubset_2 \gamma(v_2) \wedge l(v_1) = l(\gamma(v_1))$.

Next we present the net class which will be used to illustrate the main concepts and results developed in this paper. An *elementary net* is a net $N = (P, T, F)$, where P is a finite set of places, T is a finite set of transitions ($P \cap T = \emptyset$) and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation. For $x \in P \cup T$ we denote ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ (*preset* of x) and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ (*postset* of x). This notation can be extended to $X \subseteq P$ or $X \subseteq T$ by ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$. Each set $m \subseteq P$ is called a *marking*. A transition $t \in T$ is *enabled to occur* in a marking m of N if and only if ${}^\bullet t \subseteq m \wedge (m \setminus {}^\bullet t) \cap t^\bullet = \emptyset$. In this case, its occurrence leads to the marking $m' = (m \setminus {}^\bullet t) \cup t^\bullet$. Two transitions $t_1, t_2 \in T, t_1 \neq t_2$, are *in conflict* if and only if $({}^\bullet t_1 \cup t_1^\bullet) \cap ({}^\bullet t_2 \cup t_2^\bullet) \neq \emptyset$.

Definition 2.3. (Elementary net with inhibitor arcs (a-priori))
An *elementary net with inhibitor arcs* is a quadruple $ENI = (P, T, F, C_-)$, where (P, T, F) is an elementary net and $C_- \subseteq P \times T$ is the *negative context relation* (or inhibitor relation) satisfying $(F \cup F^{-1}) \cap C_- = \emptyset$. For a transition t, ${}^- t = \{p \in P \mid (p, t) \in C_-\}$ is the *negative context* of t.

A transition t is *enabled to occur* in a marking m if and only if it is enabled to occur in the underlying elementary net (P, T, F) and if ${}^- t \cap m = \emptyset$. The occurrence of an enabled transition t leads to the marking $m' = (m \setminus {}^\bullet t) \cup t^\bullet$.

Two transitions $t_1, t_2 \in T, t_1 \neq t_2$, are in *synchronous conflict* (in the a-priori semantics) if they are in conflict in the underlying elementary net or if $({}^\bullet t_i) \cap ({}^- t_j) \neq \emptyset$ (for $i, j \in \{1, 2\}, i \neq j$). A set of transitions $s \subseteq T$, called *synchronous step*, is *enabled to occur* in a marking m of ENI if and only if every $t \in s$ is enabled to occur in m and no two transitions $t_1, t_2 \in s, t_1 \neq t_2$, are in synchronous conflict. In this case, its occurrence leads to the marking $m' = (m \setminus {}^\bullet s) \cup s^\bullet$. We write $m \xrightarrow{s} m'$ to denote that s is enabled to occur in m and that its occurrence leads to m'. ○

In the example net of Figure 1 (part (a)) the negative context relation is depicted through so called inhibitor arcs with circles as arrowheads (the standard arcs represent the usual flow relation). An example for the occurrence rule of elementary nets with inhibitor arcs equipped with the a-priori semantics by means of this example net is described in the introduction.

Now we introduce the "classical" process semantics for ENI as presented in [10]. Remember that since the absence of a token in a place cannot be directly represented in an occurrence net, every inhibitor arc is replaced by a read arc to a complement place. Moreover, such complement places remove possible contact situations. These are situations, when the enabledness of a transition is prohibited by tokens in the postset of the transition. It is shown in [20] that ENI can be transformed via *complementation* into a contact-free elementary net with positive context (i.e. with read arcs depicted through arcs with dots as arrowheads) exhibiting the same behavior. The set of complement places[6] will be denoted by P' and the complementation-bijection from P to P' will be denoted by c. The processes of ENI are defined endowing processes of "ordinary" elementary nets (defined as usual by occurrence nets using complementation) with read arcs (also called activator arcs in [10, 15, 16]).

Definition 2.4. (Labelled occurrence net)
An *occurrence net* is a net $O = (B, E, R)$ satisfying:

(i) $|{}^\bullet b|, |b^\bullet| \leq 1$ for every $b \in B$ (places are *unbranched*).

(ii) O is *acyclic*, i.e. the transitive closure R^+ of R is a partial order.

Places of an occurrence net are called *conditions* and transitions of an occurrence net are called *events*.

The set of conditions of an occurrence net $O = (B, E, R)$ which are minimal (maximal) according to R^+ are denoted by $Min(O)$ ($Max(O)$).

A labelled occurrence net is a tuple (O, l) where O is an occurrence net and l is a labelling function on $B \cup E$. ○

Definition 2.5. (Process of an elementary net)
Let $N = (P, T, F)$ be an elementary net and m_0 be a marking. A *process of* N w.r.t. m_0 is a labelled occurrence net $O = (B, E, R, l)$ such that the following conditions are satisfied:

(i) No isolated place of O is mapped by l to a complement place $p \in P'$ (a place is isolated if it has an empty preset and postset).

(ii) $l|_D$ is injective for every maximal co-set D of $(B \cup E, R^+)$.

(iii) $l(Min(O)) \cap P = m_0 \wedge l(Min(O)) \subseteq m_0 \cup \{p' \in P' \mid c^{-1}(p') \notin m_0\}$.

[6] The concept of complement places can often be simplified (omitting complement places or using existing places as complement places); such principles are applied in graphical representations, e.g. see Figure 1, part (b).

(iv) $\forall e \in E : l({}^\bullet e) = {}^\bullet l(e) \cup \{p' \in P' \mid c^{-1}(p') \in l(e)^\bullet\} \;\wedge\; l(e^\bullet) = l(e)^\bullet \cup \{p' \in P' \mid c^{-1}(p') \in {}^\bullet l(e)\}$.

○

Definition 2.6. (Activator process)
A *labelled activator occurrence net* (ao-net) is a five-tuple $AON = (B, E, R, Act, l)$ satisfying: (B, E, R, l) is a labelled occurrence net, $Act \subseteq P \times T$ is the *positive context relation* satisfying $(R \cup R^{-1}) \cap Act = \emptyset$, and the relational structure

$$\mathcal{S}(AON) = (E, \prec, \sqsubset) \\ = (E, (R \circ R)|_{E \times E} \cup (R \circ Act), (Act^{-1} \circ R) \setminus id_E)$$

is $\diamondsuit$-acyclic. An ao-net AON is an *activator process* of $ENI = (P, T, F, C_-)$ w.r.t. a marking m_0 if and only if:

- $O = (B, E, R, l)$ is a process of the elementary net $N = (P, T, F)$ w.r.t. m_0, and
- $(\forall b \in B, \forall e \in E :\ (b, e) \in Act \Longrightarrow (c^{-1}(l(b)), l(e)) \in C_-)$ and $(\forall e \in E :\ (p, l(e)) \in C_- \Longrightarrow |\{b \in B \mid c^{-1}(l(b)) = p,\ (b, e) \in Act\}| = 1)$.

In this case the labelled so-structure $(\mathcal{S}(AON)^\diamondsuit, l)$ is called a run of ENI w.r.t. m_0. Denote by $\mathbf{Run}(ENI, m_0)$ the set of all runs of ENI w.r.t. m_0. ○

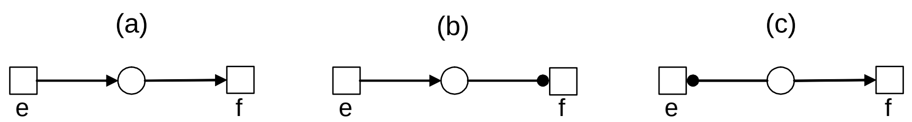

Figure 6. The nets in (a) and (b) generate the order $e \prec f$, the net in (c) the order $e \sqsubset f$.

An example of an activator process and an associated run is depicted in Figure 1 (parts (b) and (c)). The construction rule of $\mathcal{S}(AON)$ is illustrated in Figure 6. For a more detailed definition of activator processes and a discussion of related results see the series of papers [10, 15, 16].

The central idea behind the modelling of restricted occurrence rules as in the case of inhibitor nets on the algebraic level is the utilization of partial algebras [4] in the context of partial composition rules for process terms. A *partial algebra* is a set called *carrier* together with a set of (partial) operations (with possibly different arity) on the carrier. A partial algebra with one binary operation is a *partial groupoid*, i.e. an ordered tuple $\mathcal{I} = (I, dom_{\dot+}, \dot+)$, where I is the *carrier* of $\mathcal{I}$, $dom_{\dot+} \subseteq I \times I$ is the *domain* of $\dot+$, and $\dot+ :\ dom_{\dot+} \to I$ is the *partial operation* of $\mathcal{I}$.

Definition 2.7. (Partial closed commutative monoid)
A partial groupoid $\mathcal{I} = (I, dom_{\dot+}, \dot+)$ is called a *partial closed commutative monoid* if the following conditions are satisfied:

- If $a \dot+ b$ is defined then also $b \dot+ a$ is defined with $a \dot+ b = b \dot+ a$ (*closed commutativity*).

- If $(a \dotplus b) \dotplus c$ is defined then also $a \dotplus (b \dotplus c)$ is defined with $(a \dotplus b) \dotplus c = a \dotplus (b \dotplus c)$ (*closed associativity*).
- There is a (unique) *neutral element* $i \in I$ such that $a \dotplus i$ is defined for all $a \in I$ with $a \dotplus i = a$ (*existence of a (total) neutral element*).

○

We shortly recall the concept of closed congruences on partial algebras. Given a partial algebra with carrier X, an equivalence relation $\sim$ on X is called *congruence* if for each n-ary operation *op* on X with domain dom_{op}: $a_1 \sim b_1, \ldots, a_n \sim b_n$, $(a_1, \ldots, a_n) \in dom_{op}$ and $(b_1, \ldots, b_n) \in dom_{op}$ implies $op(a_1, \ldots, a_n) \sim op(b_1, \ldots, b_n)$. A congruence $\sim$ is called *closed* if for each n-ary operation *op* on X with domain dom_{op}: $a_1 \sim b_1, \ldots, a_n \sim b_n$ and $(a_1, \ldots, a_n) \in dom_{op}$ implies $(b_1, \ldots, b_n) \in dom_{op}$. Thus, a congruence is an equivalence which preserves all operations of a partial algebra. A closed congruence moreover preserves the domains of operations. Therefore the operations of a partial algebra $\mathcal{X}$ with carrier X can be carried over to the set of equivalence classes of a closed congruence $\sim$ as follows: Denote

- $[x]_\sim = \{y \in X \mid x \sim y\}$,
- $X/_\sim = \{[x]_\sim \mid x \in X\}$,
- $dom_{op/_\sim} = \{([a_1]_\sim, \ldots, [a_n]_\sim) \mid (a_1, \ldots, a_n) \in dom_{op}\}$, and
- $op/_\sim([a_1]_\sim, \ldots, [a_n]_\sim) = [op(a_1, \ldots, a_n)]_\sim$ for each n-ary operation $op : dom_{op} \to X$ of $\mathcal{X}$ (this is well defined for closed congruences).

This defines a partial algebra $\mathcal{X}/_\sim$ with carrier $X/_\sim$ and operations $op/_\sim$. $\mathcal{X}/_\sim$ is called *factor algebra* of $\mathcal{X}$ w.r.t. $\sim$. A possibility to generate (closed) congruences on partial algebras is through so called *(closed) homomorphisms* [4]. The most important result of [4] for this paper is that there always exists a unique greatest closed congruence on a given partial algebra. For more details on partial algebras see e.g. [4].

3. Algebraic $(\mathcal{M}, \mathcal{I})$-nets

"Petri nets are monoids" is the title and the central idea of the paper [19]. It provides an algebraic approach to define both nets and their processes as terms. A crucial assumption for this concept is that arbitrary concurrent composition of processes is defined, which holds true for place/transition Petri nets where places can hold arbitrarily many tokens. But for Petri nets with restricted occurrence rules like inhibitor nets or even simple elementary nets there exist transitions that cannot occur concurrently or synchronously. Therefore the respective algebraic operations will here be defined partially. Note that the synchronous composition operator was not considered in [19].

As a starting point a *general algebraic Petri net* is defined similarly as in [19] to be a quadruple $\mathcal{A} = (M, T, pre\colon T \to M, post\colon T \to M)$, where M is the set of *markings* and T is the set of *transitions*. Formally, the set of markings M is equipped with an operation $+$ such that $\mathcal{M} = (M, +)$ is a (total) commutative monoid with neutral element $\underline{0}$. The two mappings $pre\colon T \to M, post\colon T \to M$

assign *presets* and *postsets* to each transition. The behaviour of general algebraic Petri nets is given by so called *process terms* which are defined inductively: Each transition t is an *elementary* process term representing a transformation from the marking $pre(t)$ into the marking $post(t)$. Also each marking m is an elementary process term representing a "transformation" from the marking $pre(m) = m$ into the marking $post(m) = m$. To construct more complex process terms we introduce concurrency and synchronicity as well as a sequential firing rule: Firstly transitions can be synchronously composed to *synchronous step terms*, and secondly markings and synchronous step terms can be sequentially and concurrently composed to (general) *process terms*. For example, if a transition t and a marking m can be composed concurrently to the term $t \parallel m$, this term changes the marking $pre(t) + m$ to $post(t) + m$. The restricted concurrent and synchronous occurrence rule is encoded in the partiality of the respective composition operators. This partiality is modelled by equipping process terms with information elements from a partial algebra of information using the results about partial algebras from section 2. The general approach of process terms that not only ought to model the occurrence rules but also arbitrary processes is developed in the following.

Each process term α has assigned an *initial marking* $pre(\alpha) \in M$ and a *final marking* $post(\alpha) \in M$, written $\alpha : pre(\alpha) \to post(\alpha)$. Two process terms can be *sequentially composed*, if the final marking of the first process term equals the initial marking of the second process term. Moreover, each marking and each transition has assigned an information element used for determining the synchronous composability of transitions and the concurrent composability of process terms. Thus, a set of information elements I is equipped with the partial operations $\dot{\parallel} : dom_{\dot{\parallel}} \to I$ and $\dot{\oplus} : dom_{\dot{\oplus}} \to I$ for the concurrent and synchronous composition of information elements, resulting in a partial algebra $\mathcal{I} = (I, dom_{\dot{\parallel}}, \dot{\parallel}, dom_{\dot{\oplus}}, \dot{\oplus})$. The relations $dom_{\dot{\parallel}}, dom_{\dot{\oplus}} \subseteq I \times I$ specify the pairs of *(concurrently resp. synchronously) independent* information elements. The groupoids $(I, dom_{\dot{\parallel}}, \dot{\parallel})$ and $(I, dom_{\dot{\oplus}}, \dot{\oplus})$ are assumed to be partial closed commutative monoids with neutral elements i_0 and j_0. Such an $\mathcal{I}$ is called an *synchronous-concurrent partial algebra* (*sc-partial algebra*).

Definition 3.1. (Algebraic $(\mathcal{M}, \mathcal{I})$-net)
Let $\mathcal{I} = (I, dom_{\dot{\parallel}}, \dot{\parallel}, dom_{\dot{\oplus}}, \dot{\oplus})$ be an sc-partial algebra, $\mathcal{M} = (M, +)$ be a total commutative monoid, $\mathcal{A} = (M, T, pre\colon T \to M, post\colon T \to M)$ be a general algebraic Petri net, and $inf : M \cup T \to I$ be a mapping. Then $(\mathcal{A}, inf)$ is called an *algebraic $(\mathcal{M}, \mathcal{I})$-net*. ◯

Two transitions can be *synchronously composed*, if their associated information elements can be synchronously composed. Their synchronous composition yields a synchronous step term, which has associated as information element the synchronous composition of the information elements of the two transitions. This procedure can be iterated leading to synchronous step terms consisting of more than two transitions. Thus, in general the synchronous step terms of an algebraic $(\mathcal{M}, \mathcal{I})$-net are defined inductively as follows.

Definition 3.2. (Synchronous step terms)
Let $(\mathcal{A}, inf)$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net. Its *elementary synchronous step terms* are its transitions $t \in T$. If s and s' are synchronous step terms which satisfy $(inf(s), inf(s')) \in dom_{\dot{\oplus}}$, then their synchronous composition yields the *synchronous step term* $s \oplus s'$ with *initial marking* $pre(s \oplus s') = pre(s) + pre(s')$, *final marking* $post(s \oplus s') = post(s) + post(s')$ and assigned *information element* $inf(s \oplus s') = inf(s) \dot{\oplus} inf(s')$. The set of all synchronous step terms of $(\mathcal{A}, inf)$ is denoted by $Step_{(\mathcal{A}, inf)}$. ◯

Observe that this definition extends the mappings pre, $post$ and inf to synchronous step terms. Each $s \in Step_{(\mathcal{A},inf)}$ has the form $s = v_1 \oplus \ldots \oplus v_n$ for transitions $v_1, \ldots, v_n \in T$. We denote $t \in s$ if $\exists i \in \{1, \ldots, n\}$: $t = v_i$, and we define $|s| \in \mathbb{N}^T$ by $|s|(t) = |\{i \in \{1, \ldots, n\} \mid t = v_i\}|$.

Next we define the process term semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets through sequential and concurrent composition of markings and synchronous step terms. Each process term has assigned a set of information elements (information set). For markings and synchronous step terms, the associated information set will contain only the information element assigned by the mapping inf. The sequential composition of two process terms has assigned the union of their respective information sets because the information elements of both terms have to be regarded for possible further concurrent composition operations. The concurrent composition of two process terms has assigned the set of concurrent compositions of the information elements in their respective information sets. Note that the sequential composition ; as well as the concurrent composition $\|$ are partial: For sequential composability the final marking of the first process term has to coincide with the initial marking of the second process term. For concurrent composability the information sets of the two process terms have to be (concurrently) independent. Two information sets X and Y are called *(concurrently) independent* if each information element in X is (concurrently) independent from each information element in Y.

Since in general information sets instead of information elements are needed to decide the concurrent composability of process terms the sc-partial algebra $\mathcal{I} = (I, dom_{\dot{\|}}, \dot{\|}, dom_{\dot{\oplus}}, \dot{\oplus})$ of information elements is lifted to the partial algebra of information sets $\mathcal{X} = (2^I, dom_{\{\dot{\|}\}}, \{\dot{\|}\}, 2^I \times 2^I, \cup)$ defined by $dom_{\{\dot{\|}\}} = \{(X, Y) \in 2^I \times 2^I \mid X \times Y \subseteq dom_{\dot{\|}}\}$ and $X\{\dot{\|}\}Y = \{x \,\dot{\|}\, y \mid x \in X \wedge y \in Y\}$. Since only transitions will be synchronously composed, contrariwise to $\dot{\|}$ it is not necessary to consider the operation $\dot{\oplus}$ for general process terms here. For the sequential composition we have to add the total operation $\cup$ (union) on 2^I ($\cup$ is total because the sequential composition is not restricted through information sets but through pre- and postsets). It is easy to verify that $\mathcal{X}$ is also a partial closed commutative monoid [11]. Two information sets A and B can *carry the same "information"* in the sense that each information set C is either independent from both A and B or not independent from both A and B. Such sets need not be distinguished and can be technically identified through a closed congruence on 2^I. Therefore we distinguish information sets only up to the greatest closed congruence $\cong \in 2^I \times 2^I$ on $\mathcal{X}$ whose equivalence classes in our case will represent the minimal information which can be assigned to process terms. Based on these preparations process terms of an algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, inf)$ which represent all abstract computations of $(\mathcal{A}, inf)$ are defined inductively as follows:

Definition 3.3. (Process terms)
Let $(\mathcal{A}, inf)$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net. Its *elementary process terms* are of the form $id_a : a \longrightarrow a$ with $Inf(id_a) = [\{inf(a)\}]_{\cong}$ for $a \in M$ (mostly we denote id_a simply by a) and $s : pre(s) \longrightarrow post(s)$ with $Inf(s) = [\{inf(s)\}]_{\cong}$ for $s \in Step_{(\mathcal{A},inf)}$.

If $\alpha : a_1 \longrightarrow a_2$ and $\beta : b_1 \longrightarrow b_2$ are process terms satisfying $(Inf(\alpha), Inf(\beta)) \in dom_{\{\dot{\|}\}/\cong}$, their *concurrent composition* yields the process term

$$\alpha \,\|\, \beta \colon a_1 + b_1 \longrightarrow a_2 + b_2$$

with $Inf(\alpha \,\|\, \beta) = Inf(\alpha) \, \{\dot{\|}\}/_{\cong} \, Inf(\beta)$.

If $\alpha : a_1 \longrightarrow a_2$ and $\beta : b_1 \longrightarrow b_2$ are process terms satisfying $a_2 = b_1$, their *sequential composition* yields the process term

$$\alpha;\beta : a_1 \longrightarrow b_2$$

with $Inf(\alpha;\beta) = Inf(\alpha) \cup/_{\cong} Inf(\beta)$.

The *partial algebra of all process terms* with the partial operations of synchronous, concurrent and sequential composition will be denoted by $\mathcal{P}(\mathcal{A}, inf)$. ○

Note that the mappings pre and $post$ are extensions of the previously defined mappings. For a process term α we denote by B_α the set of synchronous step terms α is composed from (using some markings and the operations $\|$ and ;). The synchronous step terms in B_α are called *basic step terms of* α. The basic step terms of the process term from Figure 3 are $e \oplus f$ and $g \oplus h$. For a process term α and $t \in T$ we denote $t \in \alpha \Longleftrightarrow (\exists s \in B_\alpha : t \in s)$.

As an example of algebraic $(\mathcal{M}, \mathcal{I})$-nets and process term semantics we show now how to instantiate elementary nets with inhibitor arcs as $(\mathcal{M}, \mathcal{I})$-nets. This way algebraic process term semantics can be defined for elementary nets with inhibitor arcs.

Example 3.1. (Elementary nets with inhibitor arcs)
To instantiate an elementary net with inhibitor arcs $ENI = (P, T, F, C_-)$ (see section 2) as an algebraic $(\mathcal{M}, \mathcal{I})$-net, we have to appropriately fix the total commutative monoid of markings $\mathcal{M} = (M, +)$, an sc-partial algebra $\mathcal{I} = (I, dom_{\dot{\|}}, \dot{\|}, dom_{\dot{\oplus}}, \dot{\oplus})$ of information and the mappings pre, $post$ and inf. This instantiation will be consistent with the operational step semantics. That means, a set of transitions s is concurrently enabled in ENI at a marking m if and only if s is a defined synchronous step term of the algebraic net and there is a marking x with $pre(s) + x = m$ such that $s \| x$ is a defined proces term of the algebraic net (this relationship will be formalized on a general level in section 5). Denote [11]:

- $\mathcal{M} = (M, +) = (2^P, \cup)$, $pre(t) = {}^\bullet t$ and $post(t) = t^\bullet$, since elementary nets have sets of places as markings.
- $I = 2^P \times 2^P \times 2^P$, , $inf(t) = ({}^\bullet t, t^\bullet, {}^- t)$ $(t \in T)$ and $inf(m) = (m, m, \emptyset)$ $(m \in M)$, i.e. the assigned information distinguishes preset, postset and context information of transitions and markings in information triples. The components of such information triples we call pre-, post- and context-part.

That means any information necessary for the occurrence rule is implemented in the information elements (see section 2). For the example net from Figure 1 (part (a)) we have $inf(e) = (\{p_1\}, \{p_3\}, \emptyset)$, $inf(f) = (\{p_2\}, \{p_4\}, \{p_3\})$, $inf(g) = (\{p_5\}, \{p_6\}, \{p_7\})$, $inf(h) = (\{p_4\}, \{p_7\}, \{p_6\})$. It is now important which information triples can be composed synchronously respectively concurrently, and which information triples result from such a composition. Completely coincident with the occurrence rule of elementary nets with inhibitor arcs equipped with the a-priori semantics (see section 6 for the proofs), two information elements $i_1 = (a, b, c), i_2 = (d, e, f) \in I$ can be composed

- concurrently if and only if the pre- and post-part of i_1 is disjoint from all parts of i_2 and vice versa: $(a \cup b) \cap (d \cup e) = (a \cup b) \cap f = c \cap (d \cup e) = \emptyset$. That means $dom_{\dot{\|}} = \{((a, b, c), (d, e, f)) \mid (a \cup b) \cap (d \cup e) = (a \cup b) \cap f = c \cap (d \cup e) = \emptyset\}$. Their concurrent composition yields $(a, b, c) \dot{\|} (d, e, f) = (a \cup d, b \cup e, c \cup f)$.

- synchronously if and only if the pre- and post-parts of i_1 and i_2 are disjoint and the pre-part of i_1 is disjoint from the context-part of i_2 and vice versa: $(a \cup b) \cap (d \cup e) = a \cap f = d \cap c = \emptyset$. That means $dom_{\dot\oplus} = \{((a,b,c),(d,e,f)) \mid (a \cup b) \cap (d \cup e) = a \cap f = d \cap c = \emptyset\}$. Their synchronous composition yields $(a,b,c) \dot\oplus (d,e,f) = (a \cup d, b \cup e, (c \cup f) \setminus (b \cup e))$.

In in our running example the only pair of transitions that cannot be composed synchronously is f with h. Note that e and f as well as g and h can be composed synchronously with $inf(e \oplus f) = (\{p_1,p_2\},\{p_3,p_4\},\emptyset)$ and $inf(g \oplus h) = (\{p_4,p_5\},\{p_6,p_7\},\emptyset)$. The illustrated principle of synchronous composition can be iterated. In this way also $e \oplus f \oplus g$ and $e \oplus g \oplus h$ are defined synchronous step terms.

A possible more complex process term is $((e \oplus f) \parallel \{p_5\}); ((g \oplus h) \parallel \{p_3\})$ (see also Figure 3). The intention of this process term is to describe the behaviour of firstly firing e and f synchronously and then firing g and h synchronously. This leads directly to the left graphic of Figure 7 because $\oplus$ represents synchronicity and ; a sequential ordering (as shown above and discussed in the introduction, the respective transitions can be composed synchronously). When constructing the process term, the problem is that $pre(e \oplus f) = \{p_1,p_2\}$ does not match the initial marking $\{p_1,p_2,p_5\}$ of the net and that $post(e \oplus f) = \{p_3,p_4\}$ is not equal to the initial marking of the follower step $pre(g \oplus h) = \{p_4,p_5\}$. This problem is solved by adding (concurrent) markings comprising of the places missing in each case. These markings are added via concurrent composition leading to the next graphic in Figure 7 representing the final process term. On the right side of Figure 7 the discussed causal behaviour is developed in the form of an so-structure: First the two synchronous steps of transitions are constructed with symmetric "not later than" relations and then the two steps are arranged sequentially with an "earlier than" relation. The formal relationships of process terms to so-structures will be developed in the next section.

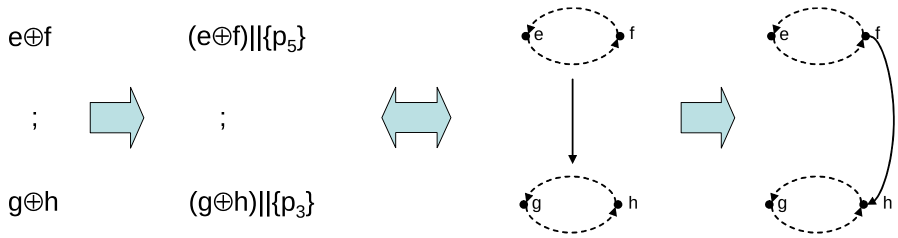

Figure 7. Informal development of a process term and an so-structure modeling the same behaviour.

Compared to [13] and [11], the definition of algebraic $(\mathcal{M},\mathcal{I})$-nets in this section is as general as possible. In order to derive conclusions about process term semantics on the algebraic level similar as in [11] it is necessary to require certain properties for the mapping inf of an algebraic $(\mathcal{M},\mathcal{I})$-net $(\mathcal{A}, inf)$, relating the sets $(\mathcal{I}, dom_{\dot\parallel}, \dot\parallel)$, $(\mathcal{I}, dom_{\dot\oplus}, \dot\oplus)$ and $\mathcal{M} = (M,+)$. All properties have a simple intuitive interpretation. For all common net classes (with so-structure based semantics) it is easy to show that they are fulfilled. In contrast to [13] where no results are obtained on the abstract level we have to introduce more specific properties for inf. We did not include them into the algebraic $(\mathcal{M},\mathcal{I})$-net definition. Instead, for each stated result we will explicitly mention which properties are required. These properties are for $x, y, m, m_1, m_2 \in M$ and $s, s_1, s_2 \in Step_{(\mathcal{A},inf)}$:

(Con1) $(inf(x), inf(y)) \in dom_{\dot\parallel} \Longrightarrow inf(x+y) = inf(x) \dot\parallel inf(y)$ (consistency of markings)

(Con2) $\mathit{inf}(\underline{0}) = i_0$ (consistency of neutral elements)

(Con3) $\{\mathit{inf}(s)\} \cong \{\mathit{inf}(s), \mathit{inf}(\mathit{pre}(s)), \mathit{inf}(\mathit{post}(s))\}$ (consistency of steps and initial/final marking)

(Con4) $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in \mathit{dom}_{\dot{\oplus}} \Longrightarrow \{\mathit{inf}(s_1 \oplus s_2)\} \cong \{\mathit{inf}(s_1 \oplus s_2), \mathit{inf}(s_1), \mathit{inf}(s_2)\}$ (consistency of steps)

(Con5) $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in \mathit{dom}_{\dot{\oplus}}$, $(\mathit{inf}(s_1) \dot{\oplus} \mathit{inf}(s_2),\ \mathit{inf}(m))$, $(\mathit{inf}(s_1), \mathit{inf}(m_1))$, $(\mathit{inf}(s_2), \mathit{inf}(m_2)) \in \mathit{dom}_{\dot{\|}}$, $\mathit{pre}(s_1) + \mathit{pre}(s_2) + m = \mathit{pre}(s_1) + m_1$, $\mathit{post}(s_1) + m_1 = \mathit{pre}(s_2) + m_2$ $\Longrightarrow (\mathit{inf}(\mathit{pre}(s_2) + m), \mathit{inf}(s_1)), (\mathit{inf}(\mathit{post}(s_1) + m), \mathit{inf}(s_2)) \in \mathit{dom}_{\dot{\|}}$ (synchronous-sequential consistency)

(Con6) $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in \mathit{dom}_{\dot{\|}} \Longrightarrow (\mathit{inf}(s_1), \mathit{inf}(s_2)) \in \mathit{dom}_{\dot{\oplus}}$ and $\{\mathit{inf}(s_1) \dot{\|}\, \mathit{inf}(s_2)\} \cong \{\mathit{inf}(s_1) \dot{\|}\, \mathit{inf}(s_2),\ \mathit{inf}(s_1) \dot{\oplus} \mathit{inf}(s_2)\}$ (synchronous consistency)

(Det) $(\mathit{inf}(s), \mathit{inf}(x)), (\mathit{inf}(s), \mathit{inf}(y)) \in \mathit{dom}_{\dot{\|}}, \mathit{pre}(s) + x = \mathit{pre}(s) + y \Longrightarrow \mathit{post}(s) + x = \mathit{post}(s) + y$ (determinism)

The first two consistency properties (Con1) and (Con2) are self explanatory. Property (Con3) states that the information (about concurrent composability) attached to a synchronous step s includes information about $\mathit{pre}(s)$ and $\mathit{post}(s)$ and (Con4) tells that it also includes information about sub-steps of s. The synchronous-sequential consistency (Con5) can be interpreted as follows: if two synchronous step terms s_1, s_2 can occur synchronously and sequentially in the order $s_1 \longrightarrow s_2$ in the same initial marking, then the occurrence of s_2 does not depend on the final marking of the occurrence of s_1 and the occurrence of s_1 does not depend on the initial marking of the occurrence of s_2. The next condition (Con6) determines that two synchronous step terms, which can occur concurrently, can also occur synchronously and that the information associated to their concurrent composition includes the information associated to their synchronous composition. For net classes we are interested in, the occurrence of a step s in a marking m is deterministic in the sense that the follower marking m' is unique (Det).

We conclude this section with a technical notion concerning the greatest closed congruence $\cong$ on the partial algebra of information sets $\mathcal{X}$. For an information set $A \in 2^I$ we abbreviate $[A] = [A]_{\cong}$. It is convenient to carry the subset relation $\subseteq$ on 2^I over to $2^I/_{\cong}$, thus defining when a congruence class $[A]$ represents less information than a congruence class $[B]$ in the following sense: if $[B]$ can be composed concurrently (using $\{\dot{\|}\}_{/\cong}$) with a congruence class $[C]$ then also $[A]$ (representing less information) can be composed concurrently with $[C]$. Therefore we define the respective relation $\subseteq_{/\cong}$ as follows:

Definition 3.4. For $a, b \in 2^I/_{\cong}$ we write $a \subseteq_{/\cong} b$ if and only if there exist $A, B \in 2^I$ with $A \subseteq B$, $a = [A]$ and $b = [B]$. ◯

Simple technical computations yield the following properties of $\subseteq_{/\cong}$:

Lemma 3.1. Let $a, b, a', b' \in 2^I/_{\cong}$:
(i) $\subseteq_{/\cong}$ is a *weak partial order*, i.e. $\subseteq_{/\cong}$ is reflexive, transitive and antisymmetric.
(ii) $a \subseteq_{/\cong} a'$, $b \subseteq_{/\cong} b'$, $a' \{\dot{\|}\}_{/\cong} b'$ defined $\Longrightarrow a \{\dot{\|}\}_{/\cong} b$ defined.
(iii) $a \subseteq_{/\cong} a'$, $b \subseteq_{/\cong} b' \Longrightarrow (a \cup_{/\cong} b) \subseteq_{/\cong} (a' \cup_{/\cong} b')$ and $(a \{\dot{\|}\}_{/\cong} b) \subseteq_{/\cong} (a' \{\dot{\|}\}_{/\cong} b')$ (if defined).
(iv) $a \subseteq_{/\cong} (a \cup_{/\cong} b)$ and $a \subseteq_{/\cong} (a \{\dot{\|}\}_{/\cong} b)$ (if defined).
(v) $a \subseteq_{/\cong} c$, $b \subseteq_{/\cong} c \Longrightarrow (a \cup_{/\cong} b) \subseteq_{/\cong} c$.

Proof:
[Sketch of the proof] Technical argumentation using the definition of the greatest closed congruence. See the Appendix for a detailed proof. □

These results directly carry over to the composability of process terms and the information attached to composed process terms. In particular, we deduce that sub-terms of a process term have associated less information than the process term, where sub-terms are as usual defined inductively following the inductive definition of process terms. The properties of $\subseteq/_{\cong}$ summarized above will be fundamentally used in the proofs of this paper without explicitly mentioning them anymore.

4. Causal semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets

In this section we explain the filled arc from process terms to runs in Figure 5 and establish results forming the basis for the coincidence of runs from process nets and runs from process terms as it is illustrated in Figure 5. First we define explicit causal semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets by associating so-structures to process terms.

Definition 4.1. (So-structures of process terms)
We define inductively *labelled so-structures* $\mathcal{S}_\alpha = (V, \prec_\alpha, \sqsubset_\alpha, l_\alpha)$ *of (or associated to) a process term* α: $\mathcal{S}_m = (\emptyset, \emptyset, \emptyset, \emptyset)$ for $m \in M$, $\mathcal{S}_t = (\{v\}, \emptyset, \emptyset, l)$ with $l(v) = t$ for $t \in T$, and $\mathcal{S}_{s_1 \oplus s_2} = (V_1 \cup V_2, \emptyset, \sqsubset_1 \cup \sqsubset_2 \cup (V_1 \times V_2) \cup (V_2 \times V_1), l_1 \cup l_2)$ for synchronous step terms $s_1, s_2 \in Step_{(\mathcal{A},inf)}$ with associated so-structures $\mathcal{S}_1 = (V_1, \emptyset, \sqsubset_1, l_1)$ and $\mathcal{S}_2 = (V_2, \emptyset, \sqsubset_2, l_2)$, where the sets of nodes V_1 and V_2 are assumed to be disjoint (what can be achieved by appropriate renaming of nodes).

Finally, given process terms α_1 and α_2 with associated so-structures $\mathcal{S}_1 = (V_1, \prec_1, \sqsubset_1, l_1)$ and $\mathcal{S}_2 = (V_2, \prec_2, \sqsubset_2, l_2)$, we define

- $\mathcal{S}_{\alpha_1 \parallel \alpha_2} = (V_1 \cup V_2, \prec_1 \cup \prec_2, \sqsubset_1 \cup \sqsubset_2, l_1 \cup l_2)$,
- $\mathcal{S}_{\alpha_1 ; \alpha_2} = (V_1 \cup V_2, \prec_1 \cup \prec_2 \cup (V_1 \times V_2), \sqsubset_1 \cup \sqsubset_2 \cup (V_1 \times V_2), l_1 \cup l_2)$,

where the sets of nodes V_1 and V_2 are again assumed to be disjoint. ○

Observe that $\mathcal{S}_\alpha = (V, \prec_\alpha, \sqsubset_\alpha, l_\alpha)$ as constructed in Definition 4.1 is indeed an so-structure. Since all labelled so-structures associated to a given process term α are isomorphic (and arbitrary labelled so-structures isomorphic to $\mathcal{S}_\alpha$ are also associated to α) we mostly distinguish labelled so-structures only up to isomorphism. It is easy to verify (by an inductive proof) that a labelled so-structure $\mathcal{S}_\alpha$ of a process term α is synchronous closed.

In the context of Figure 7 we have exemplarily discussed how to evolve a process term describing a special causal behaviour. It was also mentioned that such causal behaviour can be represented by an so-structure. Now conversely every process term defines a causal behaviour that of course can be encoded by a labelled so-structure. In the example of Figure 7 we disregarded the markings of the process term because we are only interested in the causal relationships of events (states are not explicitly regarded). Therefore we have to handle the three different parts of the process term sketched in the left graphic of Figure 7: $e \oplus f$, $g \oplus h$ and the ;-connection of the first synchronous step term to the second one. As depicted in the two right graphics of Figure 7 we introduce an event for every occurrence of a transition in the process term, i.e. we draw nodes labelled by e, f, g and h. To express the synchronous steps $e \oplus f$ and $g \oplus h$ symmetric dashed "not later than" arcs are inserted between the respective events e and f as well as g and h. As explained in the Introduction this exactly expresses a synchronous occurrence of the transitions given by the labels. At last the sequential ;-ordering of the two synchronous steps is expressed by a solid "earlier than" arc from the step consisting of e and f to that of g and h. Note that in the right illustration only one such arc (from f to h) is explicitly drawn because all the other "earlier than" relations can be deduced by (C4) of Definition 2.1. The described procedure followed Definition 4.1. We started ignoring the markings and introducing events for transitions according to the definitions of $\mathcal{S}_m$ and $\mathcal{S}_t$ in Definition 4.1. Then we inserted $\sqsubset$-relations according to the definition of $\mathcal{S}_{s_1 \oplus s_2}$ and finally proceeded with $\prec$-relations as defined for $\mathcal{S}_{\alpha_1;\alpha_2}$ in Definition 4.1.

In Figure 4 more examples of so-structures associated to process terms are shown. With the first structure (from left to right) we just demonstrated the underlying principle in detail. Note that there cannot exist a process term to which the run from Figure 1 (part (c)) is associated because this so-structure is not synchronous closed. That is why we considered its linearizations (which are always synchronous closed) in Figure 4. The fact that in this example it is actually possible to find such process terms for all of these linearizations leads to the next essential idea of Definition 4.2.

We want to deduce so-structure based semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets from their process term semantics. Easy examples show that single so-structures associated to process terms in general cannot describe each run of a Petri net (e.g. as explained the run from Figure 1, part (c), is not associated to a process term; other examples which are also valid for the partial order case include so called N-forms [11]). Consequently the set of so-structures of process terms is not expressive enough in order to directly describe the complete causal semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets. But we can derive the complete causal behaviour from the set of so-structures of process terms in a similar way as in [11] for the partial order based semantics case. This complete causal behaviour will be represented by the set of so called *enabled labelled so-structures*. For their definition we denote process terms α of the form $\alpha = (s_1 \,\|\, m_1); \ldots ; (s_n \,\|\, m_n)$ ($s_1, \ldots, s_n \in Step_{(\mathcal{A}, inf)}$, $m_1, \ldots, m_n \in M$) as *synchronous step sequence terms*. The set of all synchronous step sequence terms with initial marking m is denoted by $Stepseq_{(\mathcal{A}, inf, m)}$. It is easy to observe that so-structures associated to synchronous step sequence terms are total linear.

Definition 4.2. (Enabled labelled so-structure)
A labelled so-structure $\mathcal{S}$ is *enabled to occur* in a marking m w.r.t. an algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, inf)$, if every $\mathcal{S}' \in strat_{sos}(\mathcal{S})$ is associated to some $\beta \in Stepseq_{(\mathcal{A}, inf, m)}$ (Definition 4.1). Denote by $\mathbf{Enabled}(\mathcal{A}, inf, m)$ the set of labelled so-structures enabled to occur in m w.r.t. $(\mathcal{A}, inf)$. ○

In this definition enabled labelled so-structures are introduced using linearizations. Figure 4 gives an example how to check if an so-structure is enabled. It shows that the run from Figure 1 (part (c)) is enabled w.r.t. the marked net in the same figure. We will show in Theorem 4.1, that so-structures of process terms are enabled in the initial marking of the process term. Obviously, every extension of an so-structure enabled in m is also enabled in m, because extensions have less linearizations.

A labelled so-structure $\mathcal{S}$ enabled in m is said to be *minimal*, if there exists no labelled so-structure $\mathcal{S}' \neq \mathcal{S}$ enabled in m, where $\mathcal{S}$ is an extension of $\mathcal{S}'$. We denote by $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$ the set of all such minimal enabled labelled so-structures. For example, one can check (intuitively and technically) that the run from Figure 1 (part (c)) is in $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$.

In the next definition process terms are identified through an equivalence relation. The basic idea is to identify two enabled so-structures if one is an extension of the other. Carrying over this principle to process terms we will show in Theorem 4.2 that two process terms are equivalent if their associated so-structures can be identified in the above sense. In this context the process terms in Figure 4 should all be equivalent. For algebraic $(\mathcal{M}, \mathcal{I})$-nets representing concrete Petri nets equivalent process terms will represent the same commutative process of the Petri net (for details and examples to commutative processes see [11] and [2]) in the sense that process terms are shown to be equivalent if and only if their associated so-structures are extensions of the same unique commutative run[7]. In the example all process terms in Figure 4 represent the (commutative) process in Figure 1, part (b). Note that for elementary nets (with inhibitor arcs) we do not have to distinguish between collective and individual token semantics (see [11]) and thus commutative processes and classical processes coincide.

The Theorems 4.1 and 4.2 (stated in the following of this section) on the general algebraic level are central for establishing the results illustrated in Figure 5 for concrete net classes. According to this graphic, the relation between classical process semantics and uniform algebraic process term semantics is given indirectly via associated so-structures. While the connection between processes and their associated so-structures is well known for concrete net classes, we here establish detailed results about the connection between process terms and their associated so-structures on the general level. These results form the basis for deriving the correspondence of causal semantics derived from processes and from process terms for concrete net classes.

Definition 4.3. (The congruence $\sim$)
The relation $\sim$ on the set of all process terms of an algebraic $(\mathcal{M}, \mathcal{I})$-net is the least congruence of the partial algebra of all process terms with the partial operations $\oplus$, $\parallel$ and ;[8], which includes the relation given by the following axioms for process terms $\alpha, \beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ and markings m, n:
(1) $\alpha \parallel \beta \sim \beta \parallel \alpha$
(2) $(\alpha \parallel \beta) \parallel \gamma \sim \alpha \parallel (\beta \parallel \gamma)$
(3) $(\alpha; \beta); \gamma \sim \alpha; (\beta; \gamma)$
(4) $\alpha = ((\alpha_1 \parallel \alpha_2); (\alpha_3 \parallel \alpha_4)) \sim \beta = ((\alpha_1; \alpha_3) \parallel (\alpha_2; \alpha_4))$
(5) $\alpha \oplus \beta \sim \beta \oplus \alpha$
(6) $(\alpha \oplus \beta) \oplus \gamma \sim \alpha \oplus (\beta \oplus \gamma)$
(7) $(\alpha \oplus \beta) \sim (\alpha \parallel pre(\beta)); (post(\alpha) \parallel \beta)$
(8) $(\alpha; post(\alpha)) \sim \alpha \sim (pre(\alpha); \alpha)$
(9) $id_{(m+n)} \sim id_m \parallel id_n$
(10) $pre(\alpha) + m = pre(\alpha) + n, post(\alpha) + m = post(\alpha) + n \Longrightarrow (\alpha \parallel id_m) \sim (\alpha \parallel id_n)$
(11) $(\alpha \parallel id_{\underline{0}}) \sim \alpha$
if the terms on both sides of $\sim$ are defined process terms. ○

After some explanations to this last central definition of the section the technical part of the paper will follow. The results of this technical part will complete the development of the semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets.

As discussed before the first two process terms in Figure 4 represent the same process and should consequently be equivalent (this will be proven in Theorem 4.2). Exemplarily this is shown with the

[7] A commutative run is an equivalence class of so-structures associated to a commutative process. An extension is an so-structure that extends one so-structure of the equivalence class.
[8] According to [4] this least congruence exists uniquely.

following transformation:

$((e \oplus f) \parallel \{p_5\}); ((g \oplus h) \parallel \{p_3\}) \overset{(7)}{\sim}$
$(((f \parallel \{p_1\}); (e \parallel \{p_4\})) \parallel \{p_5\}); ((g \oplus h) \parallel \{p_3\}) \overset{(8)}{\sim}$
$(((f \parallel \{p_1\}); (e \parallel \{p_4\})) \parallel (\{p_5\}; \{p_5\})); ((g \oplus h) \parallel \{p_3\}) \overset{(4)}{\sim}$
$((f \parallel \{p_1\} \parallel \{p_5\}); (e \parallel \{p_4\} \parallel \{p_5\})); ((g \oplus h) \parallel \{p_3\}) \overset{(9)}{\sim}$
$((f \parallel \{p_1, p_5\}); (e \parallel \{p_4, p_5\})); ((g \oplus h) \parallel \{p_3\})$
(note that all occurring process terms are defined w.r.t. the rules from section 3).

Given two $\sim$-equivalent process terms α and β, there holds $pre(\alpha) = pre(\beta)$ and $post(\alpha) = post(\beta)$. The so-structures associated to process terms are changed only through the axioms (4) and (7).

Regarding (4) we get that $\mathcal{S}_\alpha$ is an extension of $\mathcal{S}_\beta$ with additional $\prec$-ordering between events of α_1 and α_4 as well as between events of α_2 and α_3. Figure 8, left box, shows an example of (4) for the special case of two concurrent events g and e. A concrete example for such a situation are the concurrent transitions g and e in the running example (Figure 1). Of course such concurrent transitions can occur sequentially in any order and thus in particular in the order $g \to e$. Consequently the process term $g \parallel e$ representing the concurrent occurrence of g and h as well as the process term $(g \parallel pre(e)); (e \parallel post(g))$ representing the sequential occurrence in the order $g \to e$ are defined. The latter process term together with the respective so-structure is depicted in the right part of the box. Adding some irrelevant markings using axiom (8) the concurrent composition $g \parallel e$ can also be represented by the term $(g; post(g)) \parallel (pre(e); e)$. This term together with the associated so-structure representing the concurrency of g and e is illustrated in the left part of the box. Now axiom (4) states that, if both terms are defined, they are equivalent. This is in accord with our previous considerations because the second so-structure is an extension of the first one. That means in particular that the first so-structure is extended by both so-structures (illustrated with the $\supseteq$-symbol in the broad arc pointing at a copy of the first so-structure in the line below). Moreover this so-structure is obviously enabled. This justifies the equivalence of the two terms. A run represented by the respective equivalence class of process terms is in this example given by the so-structure below the broad arc representing the concurrent occurrence of g and e. Note that in order to formally represent a run of the net from Figure 1, part (a), the initial markings of the process terms have to coincide with the initial marking of the net. Thus by adding concurrently the marking $\{p_2\}$ to both terms, they truly represent the concurrent occurrence of g and e in this net.

Regarding (7) none of the so-structures associated to the two equivalent terms is an extension of the other one. The right box in Figure 8 shows the respective terms $e \oplus f$ and $(f \parallel pre(e)); (post(f) \parallel e)$ as well as the associated so-structures. The first term represents the synchronous occurrence of e and f while the second one represents the occurrence sequence $f \to e$. Considering the net from Figure 1, part (a), both terms are defined. An enabled so-structure extended by both so-structures associated to the process terms is given in the line below the broad arc. It describes the relationship f "not later than" e. The existence of such enabled so-structure again justifies the equivalence of the two terms. In this example this enabled so-structure is the run represented by the respective equivalence class of process terms. In order to show the general principle we have, as in the example for (4), neglected to concurrently compose the residual marking $\{p_5\}$ to the two terms.

Further observations concerning the relationship between the above axioms, the properties (Con1)-(Con6) and (Det), and the information associated to process terms are summarized in the following: In

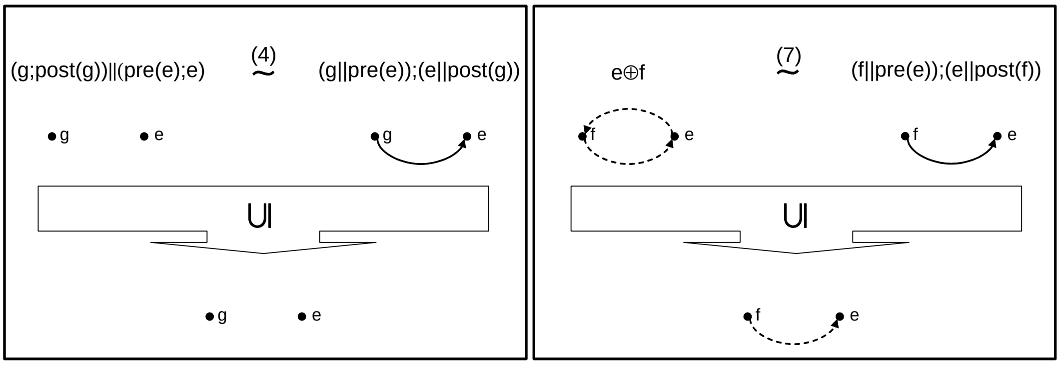

Figure 8. Illustration of the key axioms (4) and (7) from Definition 4.3.

the axioms (1), (2), (3), (5), (6), (8) one side of $\sim$ is a defined process term if and only if also the other side is. In the axioms (4) and (9) the left side is defined if the right side is defined (but not vice versa). In (11) the right side is defined if the left side is defined (and with (Con2) we have the reverse implication, too). In (7) and (10) we can derive no similar relation. In the axioms (1), (2), (3), (5) and (6) both sides have the same associated information. In (4) we have $Inf(\alpha) \subseteq/\cong Inf(\beta)$. For (7) and (10) we get no similar result. In some axioms the information of both sides is equal if adequate conditions are satisfied: In (8) we need to require (Con3) and (Con1), in (9) solely (Con1) and (Con2) in (11).

In the examples of Figure 8, we discussed the specially interesting equivalence transformations (4) and (7). The explained principles can in particular be used for sequentialization and synchronization of concurrently composed process terms. More precisely, concurrent events can occur synchronously and sequentially in any order. At the process term level this means that concurrently composed process terms can equivalently be transformed into a synchronous (only in the case of synchronous step terms) and a sequential composition of the terms. These two important transformations need a more detailed technical consideration. In order to sequentialize concurrently composed process terms we use axiom (4) with $\alpha_3 = post(\alpha_1)$ and $\alpha_2 = pre(\alpha_4)$ (similar as in the example of Figure 8). To transform concurrently composed step terms into synchronously composed step terms we additionally need axiom (7) (considering the example of Figure 8, one can first use axiom (4) and then (7) to derive a synchronous composition from a concurrent composition, if all terms are defined). For these two kinds of transformations one has to regard also axiom (8) and adequate consistency conditions. We are especially interested in the following two special cases:

- With (Con3) and (Con1) we deduce $(\alpha \parallel \beta) \sim (\alpha \parallel pre(\beta)); (\beta \parallel post(\alpha))$ and $Inf((\alpha \parallel pre(\beta)); (\beta \parallel post(\alpha))) \subseteq/\cong Inf(\alpha \parallel \beta)$.

- If additionally (Con6) is fulfilled we get $\alpha \parallel \beta \sim \alpha \oplus \beta$ and $Inf(\alpha \oplus \beta) \subseteq/\cong Inf(\alpha \parallel \beta))$.

These results will be used in equivalence transformations in the proofs of the main results mostly without mentioning them explicitly. If the associated information and so-structures stay the same we often even do not distinguish between equivalent terms anymore. In particular, we write that *a process term has without loss of generality a special form* if the process term can be equivalently transformed into that special form by such easy equivalence transformations.

In order to simplify the identification of transitions of a process term and nodes (events) of an associated so-structure it would be helpful to assume that the labelling function of an so-structure $\mathcal{S}_\alpha$ of α is the id-function. In such a case a transition would occur only once in a process term and consequently a basic step term would also occur only once in a process term. Moreover, the basic step terms of process terms could be identified with the synchronous classes of the synchronous closed so-structure $\mathcal{S}_\alpha$ since $\{|s| \mid s \in B_\alpha\} = V|_{\mathcal{S}_\alpha}$ (see section 2 for the definition of $V|_{\mathcal{S}_\alpha}$). The synchronous class corresponding to $s \in B_\alpha$ is denoted by $k_s \in V|_{\mathcal{S}_\alpha}$. To achieve this simplification for a given process term, we will identify copies of transitions of the process term with events of the associated so-structure, i.e. we define a *copy net*, where the set of transitions (the copies of the original transitions) equals the set of events of the so-structure. A copy-net may be understood as an unfolding of an algebraic net.

Definition 4.4. **((V, l)-copy net)**
Let $(\mathcal{A}, \mathit{inf})$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net with $\mathcal{A} = (M, T, \mathit{pre}, \mathit{post})$, V be a set and $l : V \to T$ be a surjective labelling function. Denote by $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ the algebraic $(\mathcal{M}, \mathcal{I})$-net given by $\mathcal{A}_{(V,l)} = (M, V, \mathit{pre}_{(V,l)}, \mathit{post}_{(V,l)})$ and $\mathit{inf}_{(V,l)} : M \cup V \to I$, where $\mathit{pre}_{(V,l)}(v) = \mathit{pre}(l(v))$, $\mathit{post}_{(V,l)}(v) = \mathit{post}(l(v))$, $\mathit{inf}_{(V,l)}|_M = \mathit{inf}|_M$ and $\mathit{inf}_{(V,l)}(v) = \mathit{inf}(l(v))$ for every $v \in V$. $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ is called *(V, l)-copy net of $(\mathcal{A}, \mathit{inf})$*. ○

The following definition is only a technicality used in the proof of Theorem 4.1 and the precursory Lemma 4.1 and Corollary 4.1. It defines the substitution of basic step terms in a given process term α. For an arbitrary set X, a set $C \subseteq B_\alpha$ and a mapping $su : C \to X$ we define inductively the *substituted term* α_{su} of α w.r.t. su: If $\alpha = m \in M$, then $\alpha_{su} = m$. If $\alpha \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$, then $\alpha_{su} = \alpha$ if $\alpha \notin C$ and $\alpha_{su} = su(\alpha)$ if $\alpha \in C$. Finally, for process terms α, β, γ: If $\alpha = \beta; \gamma$, then $\alpha_{su} = \beta_{su}; \gamma_{su}$ and if $\alpha = \beta \parallel \gamma$, then $\alpha_{su} = \beta_{su} \parallel \gamma_{su}$. In the following we will be interested in the case that basic step terms are substituted by their postsets.

In the next lemma this technique is applied to a process term α with associated so-structure $\mathcal{S}_\alpha = (V, \prec, \sqsubset, id)$ (i.e. the labelling function of the so-structure is id). That means the statement of the next lemma is formulated for process terms in which every transition occurs only once and consequently every basic step term occurs only once. Therefore the lemma is strongly connected to the technique of copy nets. It states that in such process terms minimal basic step terms can be detached to a concurrent step of these basic step terms using an equivalence transformation. Figure 9 depicts an example of this procedure: On the left side the process term $\alpha = (((t_1 \oplus t_2) \parallel m_1); (t_3 \parallel m_2)) \parallel ((t_4 \parallel m_3); ((t_5 \oplus t_6) \parallel m_4))$ is given. This process term represents the typical situation for an application of the lemma. Two sequences of synchronous step terms are concurrently composed. The respective so-structure and the associated partial order is depicted in the box below α. With the partial order we can identify the minimal basic step terms $\{t_1 \oplus t_2, t_4\}$ of α by searching for minimal nodes (synchronous classes) in the partial order. Now choosing $C = \{t_1 \oplus t_2, t_4\}$ as the whole set of minimal basic step terms, we intend to detach these basic step terms concurrently at the beginning of the process term. That means the process term is equivalently transformed to the term $\alpha_C = ((\|_{s \in C}\, s) \parallel m_C); \alpha_{su} = ((t_1 \oplus t_2 \parallel t_4) \parallel m_C); \alpha_{su}$ on the right side of Figure 9. The first part of α_C is the concurrent composition of the synchronous step terms in C together with some concurrently composed marking. A second residual part determined by the principle of substitution is sequentially composed after this first part. The effect on the associated so-structure and partial order is depicted in the box below α_C in Figure 9. The minimal events respectively basic step terms of C now have an "earlier than" ordering to all events respectively basic step terms not in C. That means sequential "earlier than" orderings separating C from the other synchronous step terms are added while the concurrency between the synchronous step terms in C is preserved. Since the resulting process term α_C has a stronger ordering than the original term α, it can be shown that α_C has attached less information than α in the sense defined by $\subseteq/_{\cong}$. In Corollary 4.1 it is shown that analogously to the next lemma the minimal basic step terms C can be detached to a synchronous step instead of a concurrent step.

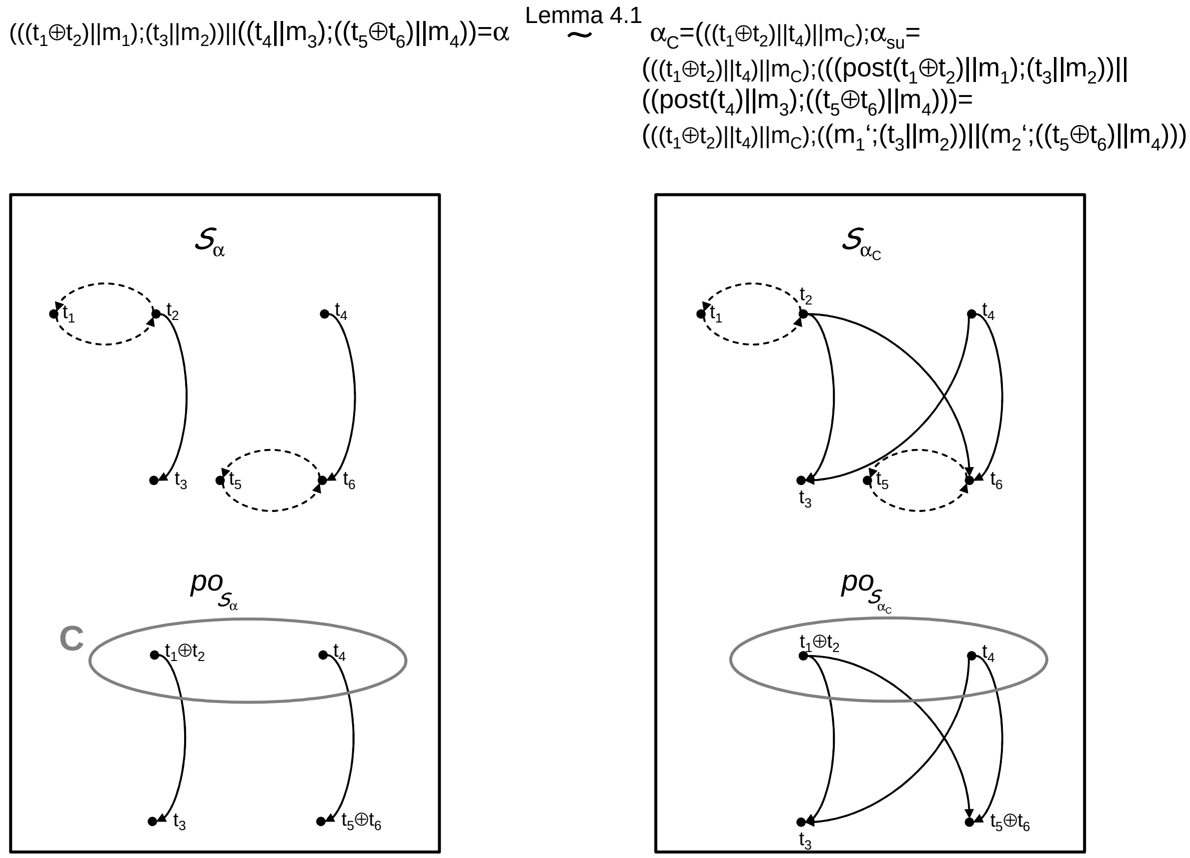

Figure 9. Illustration of (the simplest typical case of) Lemma 4.1 with $C = \{t_1 \oplus t_2, t_4\}$.

Lemma 4.1. Let α be a process term of an algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, \mathit{inf})$ which fulfills (Con1)-(Con3), such that α has no sub-term of the form $m; \alpha'$ with $m \in M$. Let further $\mathcal{S}_\alpha = (V, \prec, \sqsubset, id)$ be a labelled so-structure of α with associated partial order $po_{\mathcal{S}_\alpha} = (X|_{\mathcal{S}_\alpha}, <_{\mathcal{S}_\alpha})$ and $C \subseteq B_\alpha$ be a set of minimal basic step terms of α (i.e. k_s is minimal w.r.t. $<_{\mathcal{S}_\alpha}$ for every $s \in C$)[9]. Finally let $su : C \to M$ be given by $su(\beta) = post(\beta)$ for $\beta \in C$.

Then there exists a marking m_C, such that $\alpha_C = ((\|_{s \in C}\, s) \parallel m_C); \alpha_{su}$ is a defined process term of $(\mathcal{A}, \mathit{inf})$[10] satisfying:

(I) $\alpha_C \sim \alpha$.

[9]Corresponding synchronous classes k_s of basic step terms s are defined because the labelling function of $\mathcal{S}_\alpha$ is id.
[10]Note: α_{su} is the substituted term of α w.r.t. su.

(II) $\mathit{Inf}(\alpha_C) \subseteq/_{\cong} \mathit{Inf}(\alpha)$.

Proof:
[Sketch of the proof] Straightforward following the inductive definition of process terms (see the Appendix for a detailed proof). □

Applying the explained principles for synchronization of concurrency in process terms, α_C from Lemma 4.1 can equivalently be transformed to α_{Sy} in Corollary 4.1 (note: $m_{Sy} = m_C$). Consequently the following corollary holds:

Corollary 4.1. Assume the same preconditions as in Lemma 4.1 and additionally that $(\mathcal{A}, \mathit{inf})$ fulfills (Con6).

Then there exists a marking m_{Sy}, such that $\alpha_{Sy} = ((\oplus_{s\in C}\, s) \parallel m_{Sy}); \alpha_{su}$ is a defined process term satisfying:

(I) $\alpha_{Sy} \sim \alpha$.

(II) $\mathit{Inf}(\alpha_{Sy}) \subseteq/_{\cong} \mathit{Inf}(\alpha)$.

Now we are prepared to prove the first important theorem which shows that so-structures of process terms are enabled in the initial marking of the process term.

Theorem 4.1. Let α be a process term of an algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, \mathit{inf})$ which fulfills (Con1)-(Con3) and (Con6). Then $\mathcal{S}_\alpha \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ with $m = pre(\alpha)$. In particular, every $\mathcal{S}' \in strat_{sos}(\mathcal{S}_\alpha)$ is associated to some $\beta \in Stepseq_{(\mathcal{A},\mathit{inf},m)}$ satisfying $\alpha \sim \beta$.

Proof:
[Sketch of the proof] It is enough to construct a process term $\beta \in Stepseq_{(\mathcal{A},\mathit{inf},m)}$ which has associated a given $\mathcal{S}' \in strat_{sos}(\mathcal{S}_\alpha)$ and satisfies $\alpha \sim \beta$. Such β can be constructed by an iterative application of Corollary 4.1. See the Appendix for a detailed proof. □

An enabled so-structure $\mathcal{S}$ is uniquely determined by the set of process terms whose associated so-structures extend $\mathcal{S}$. As we have already seen in the running example the run (an enabled so-structure) from Figure 1 (part (c)) can be reconstructed from the linearizations from Figure 4 which are all associated to certain process terms.

Definition 4.5. Let $\mathcal{S} = (V, \prec, \sqsubset, l) \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. Then the set $\Upsilon_{\mathcal{S}}^{can}$ of all process terms α of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ with $pre(\alpha) = m$ whose associated so-structures extend $(V, \prec, \sqsubset, id)$ is called the *canonical set* of $\mathcal{S}$. ◯

Remark 4.1. Let $\mathcal{S} = (V, \prec, \sqsubset, l) \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ and $\mathcal{S}_\alpha = (V, \prec_\alpha, \sqsubset_\alpha, id)$ for $\alpha \in \Upsilon_{\mathcal{S}}^{can}$. Then $\mathcal{S} = (V, \bigcap_{\alpha\in\Upsilon_{\mathcal{S}}^{can}} \prec_\alpha, \bigcap_{\alpha\in\Upsilon_{\mathcal{S}}^{can}} \sqsubset_\alpha, l)$ by Proposition 2.1.

According to Definition 4.5 the set $\Upsilon_{\mathcal{S}}^{can}$ is maximal with the above property in the sense that for any set Υ of process terms of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ with initial marking m which also fulfills $\mathcal{S} = (V, \bigcap_{\alpha\in\Upsilon} \prec_\alpha, \bigcap_{\alpha\in\Upsilon} \sqsubset_\alpha, l)$ there holds $\Upsilon \subseteq \Upsilon_{\mathcal{S}}^{can}$.

The following lemma states that process terms with the same initial marking, whose associated so-structures are total linear and are all extensions of one enabled so-structure, are $\sim$-equivalent. In the subsequent theorem we can generalize this result omitting the presumption of total linear so-structures.

Lemma 4.2. Let $(\mathcal{A}, \mathit{inf})$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net fulfilling (Det) and (Con1)-(Con5). Let $\mathcal{S}'$ and $\mathcal{S}''$ be total linear labelled so-structures of process terms α and β with initial marking m. If $\mathcal{S}'$ and $\mathcal{S}''$ are extensions of $\mathcal{S} \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$, then there holds $\alpha \sim \beta$.

Proof:
[Sketch of the Proof] We show that both α and β can be equivalently transformed to one synchronous step sequence terms (which only depends on $\mathcal{S}$). Namely, it is shown that the minimal events of $\mathcal{S}$ can be equivalently permuted to the first position of a synchronous step sequence term and that this procedure can be iterated for the following events of $\mathcal{S}$. See the Appendix for a detailed proof. □

Theorem 4.2. Let $(\mathcal{A}, \mathit{inf})$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net with the same preconditions as in Lemma 4.2. Let $\mathcal{S}'$ and $\mathcal{S}''$ be labelled so-structures of process terms α and β with initial marking m. If $\mathcal{S}'$ and $\mathcal{S}''$ are extensions of $\mathcal{S} \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$, then there holds $\alpha \sim \beta$.

Proof:
There exist total linear so-structures $\mathcal{S}'_E$ respectively $\mathcal{S}''_E$ which are extensions of $\mathcal{S}'$ respectively $\mathcal{S}''$. According to Theorem 4.1 there exist synchronous step sequence terms α' resp. β' with associated so-structures $\mathcal{S}'_E$ resp. $\mathcal{S}''_E$ with $\alpha' \sim \alpha$ and $\beta' \sim \beta$. Clearly, $\mathcal{S}'_E$ and $\mathcal{S}''_E$ are also extensions of $\mathcal{S}$. Thus, according to Lemma 4.2, we get $\alpha' \sim \beta'$. Consequently we have $\alpha \sim \beta$. □

With this theorem we can identify minimal enabled so-structures through their canonical sets (use Remark 4.1).

Corollary 4.2. Let $(\mathcal{A}, \mathit{inf})$ be an algebraic $(\mathcal{M}, \mathcal{I})$-net with the same preconditions as in Theorem 4.2 and let $\mathcal{S} \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. Then $\Upsilon^{can}_{\mathcal{S}} \subseteq [\alpha]_\sim$ for some process term α of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$. If $\Upsilon^{can}_{\mathcal{S}} = [\alpha]_\sim$, then $\mathcal{S} \in \mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$.

Proof:
The first statement follows directly from Theorem 4.2, so we only prove the additional statement. For an so-structure $\mathcal{S}'$ enabled in m such that $\mathcal{S}$ is an extension of $\mathcal{S}'$ we deduce $\Upsilon^{can}_{\mathcal{S}} \subseteq \Upsilon^{can}_{\mathcal{S}'}$ by definition. From $\Upsilon^{can}_{\mathcal{S}} = [\alpha]_\sim$ it follows that $\Upsilon^{can}_{\mathcal{S}'} = [\alpha]_\sim = \Upsilon^{can}_{\mathcal{S}}$. Consequently $\mathcal{S} = \mathcal{S}'$ by Remark 4.1. □

Now we have collected all results that we need on the general level of algebraic $(\mathcal{M}, \mathcal{I})$-nets. For concrete net classes we use these results to show the relations in Figure 5. An especially interesting issue in this context is the construction principle of runs within the algebraic approach. Concluding the section this principle can be informally sketched in our running example as follows: Translate the given Petri net - here the one from Figure 1 (part (a)) - into an algebraic $(\mathcal{M}, \mathcal{I})$-net. The only problem in this step is defining an appropriate $\mathcal{I}$, i.e. $\mathcal{I}$ should encode the non-sequential occurrence rule as sketched in the previous section. Next deduce process terms: in the example we have deduced the four process terms in Figure 4. As in Figure 4 we associate so-structures to process terms. On the algebraic level we now have to identify equivalence classes according to Definition 4.3 leading to "algebraic" runs (the four equivalent process terms in Figure 4 represent the run from Figure 1, part (c)). On the causal level we have to intersect adequate so-structures of process terms in order to generate $\mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. Intersecting the so-structures in Figure 4 results in the run from Figure 1 (part (c)).

Before discussing in detail the concrete example of elementary nets with inhibitor arcs in section 6 we introduce the general framework for the instantiation of net classes as general algebraic nets in the next section.

5. The corresponding algebraic $(\mathcal{M}, \mathcal{I})$-net

In this section we present a truly general approach to construct algebraic process term semantics and subsequently causal semantics of Petri nets with restricted occurrence rule. This approach is based on the notion of algebraic $(\mathcal{M}, \mathcal{I})$-nets corresponding to concrete Petri nets.

Despite the differences between different classes of Petri nets, there are some common features shared by almost all net classes, such as the notions of *marking* (state), *transition*, and *occurrence rule* (see [6]). Thus, in the next definition we suppose a Petri net be given by a set of markings, a set of transitions and an occurrence rule determining whether a synchronous step (a multi-set) of transitions is enabled to occur in a given marking and if yes determining the follower marking. Note that for the net classes we will consider their concurrent behaviour can be obtained from the sequential and synchronous behaviour as follows: A multi-set of synchronous steps $S \in \mathbb{N}^{\mathbb{N}^T}$ is enabled to occur concurrently in a marking m if and only if S can occur synchronously and sequentially in any order in the marking m. The occurrence rule of a Petri net with a set of transitions T and a set of markings M can always be described by a transition system. Accordingly, we suppose that a Petri net is given in the form of a transition system $(M, E, \mathbb{N}^T)$ with nodes $m \in M$, labelled arcs $e \in E \subseteq M \times \mathbb{N}^T \times M$ and labels $s \in \mathbb{N}^T$, where s is interpreted as a synchronous step of transitions. The notation $m \stackrel{s}{\longrightarrow} m'$ for $(m, s, m') \in E$ means that s can occur in m with follower marking m'. The notation $m_0 \stackrel{s_1 \dots s_n}{\longrightarrow} m_n$ means that there exist $m_1, \dots, m_{n-1} \in M$, such that $m_0 \stackrel{s_1}{\longrightarrow} m_1, \dots, m_{n-1} \stackrel{s_n}{\longrightarrow} m_n$.

Definition 5.1. (Corresponding net)
Let $N = (M, E, \mathbb{N}^T)$ be a Petri net in the form of a transition system. An algebraic $(\mathcal{M}, \mathcal{I})$-net $((M, T, \mathit{pre}\colon T \to M, \mathit{post}\colon T \to M), \mathit{inf}) = (\mathcal{A}, \mathit{inf})$ is called a *corresponding net to* N if the occurrence rule for synchronous steps is preserved: if for every pair of markings $m, m' \in M$ and every synchronous step $s \in \mathbb{N}^T$ there holds $m \stackrel{s}{\longrightarrow} m'$ if and only if there exists $\tilde{s} \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}$ with $|\tilde{s}| = s$ and a marking $\tilde{m} \in M$ such that $\alpha = \tilde{s} \parallel \tilde{m}$ is a defined process term fulfilling $\mathit{pre}(\alpha) = m$ and $\mathit{post}(\alpha) = m'$. ◯

From the definitions we conclude: $(m \stackrel{s_1 \dots s_n}{\longrightarrow} m') \Longleftrightarrow$ (there exists $\alpha : m \to m' \in \mathit{Stepseq}_{(\mathcal{A}, \mathit{inf}, m)}$ of the form $\alpha = \tilde{s_1} \parallel \tilde{m_1}; \dots; \tilde{s_n} \parallel \tilde{m_n}$), where $\tilde{m_i} \in M$ and $\tilde{s_i} \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}$ with $|\tilde{s_i}| = s_i$ for every $i \in \{1, \dots, n\}$. Such α is called *corresponding to* $m \stackrel{s_1 \dots s_n}{\longrightarrow} m'$. Moreover, an so-structure associated to α is called *associated to* $m \stackrel{s_1 \dots s_n}{\longrightarrow} m'$. Altogether this describes the consistency of the algebraic approach to operational step semantics.

The construction of a corresponding algebraic $(\mathcal{M}, \mathcal{I})$-nets provides a general framework to derive causal semantics for a wide range of concrete net classes (compare the filled arcs in Figure 5). This is illustrated in section 6 for the example of elementary nets with inhibitor arcs equipped with the a-priori semantics (the respective ideas were already developed in the previous sections) using the following general scenario:

(1) Give the classical definition of a Petri net class including their synchronous step occurrence rule.

(2) Given a net N of the considered class, construct a corresponding algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, \mathit{inf})$ through defining $\mathcal{M}$, $\mathcal{I}$, pre, $post$, inf. Then deduce the partial algebra of information sets $\mathcal{X}$, the respective greatest closed congruence $\cong$ as well as a partial algebra of information isomorphic to $\mathcal{X}/_{\cong}$.

(3) Show that $(\mathcal{A}, \mathit{inf})$ satisfies the stated properties of the mapping inf (thus ensuring the validity of the theorems of section 4).

(4) Now one can derive algebraic semantics of $(\mathcal{A}, \mathit{inf})$ through process terms and thus causal semantics of N through $\mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$.

From the considerations of this section we can conclude that the causal semantics of N derived with this scheme are consistent with the operational semantics of N, because obviously (using theorem 4.1): $\mathcal{S}$ is associated to $m \stackrel{s_1 \ldots s_n}{\longrightarrow} m' \Longleftrightarrow \mathcal{S} \in strat_{sos}(\mathcal{S}')$ for some $\mathcal{S}' \in \mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$. Moreover so-structures which are not enabled never fulfill such a property and thus minimal enabled so-structures are the so-structures with the least causalities guaranteing consistency to the operational occurrence rule. These characteristics ensure that the derived causal semantics are reasonable. Consequently, if there exist non-sequential semantics of the considered Petri net class based on processes and occurrence nets, it should always be possible to show that the set of (minimal) runs representing (minimal) processes coincides with $\mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$. Moreover if there are no non-sequential semantics based on processes for a given Petri net class, they can be straightforwardly given (following the scenario above) by $\mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$. Taking a closer look on the algebraic semantics, the equivalence rules of Definition 4.3 ought to ensure the following correspondence between process terms and classical process nets: An equivalence class of process terms exactly coincides with a commutative process (given a fixed initial marking).

Altogether, the latter comments describe the respective relations of Figure 5 in more detail so that the complete framework sketched in this illustration is now revealed in all respects. All relationships that have to be implemented for concrete net classes and can only be prepared on the abstract level were already proven for many different net classes in the special case not distinguishing concurrent and synchronous behaviour in [11]. An exemplary procedure to show the consistencies of our approach to classical process semantics is presented in the next section for elementary nets with inhibitor arcs equipped with the a-priori semantics (extending the approach of [11] which is not suitable for this net class).

6. Elementary nets with inhibitor arcs

In this section we will now apply the techniques developed in the previous sections to the concrete net class of elementary nets with inhibitor arcs equipped with the a-priori semantics. Some of the main ideas, e.g. the definition of a corresponding algebraic $(\mathcal{M}, \mathcal{I})$-net, were already partially discussed on the basis of the running example net in Figure 1 (part (a)). Note that the content of this section is based on the process semantics introduced by Janicki and Koutny (see section 2). Similar results as in this section have been derived in [13]. But in [13] there was only shown a one to one correspondence between the process term semantics and the process semantics in a complicated lengthy ad-hoc way without regarding causal behaviour. Here we additionally get the complete consistency of the causal behaviour derived from process terms and the causality of activator processes.

Given an elementary net with inhibitor arcs $ENI = (P, T, F, C_-)$ (see section 2) we construct a corresponding algebraic $(\mathcal{M}, \mathcal{I})$-net analogously as in [11] by (see also section 3)

- $\mathcal{M} = (2^P, \cup)$, $I = 2^P \times 2^P \times 2^P$, $pre(t) = {}^\bullet t$, $post(t) = t^\bullet$, $inf(t) = ({}^\bullet t, t^\bullet, {}^- t)$ $(t \in T)$ and $inf(m) = (m, m, \emptyset)$ $(m \in M)$

distinguishing preset, postset and context information (of transitions) in information triples. Therefore we can encode the occurrence rule by the following partial operations:

- $dom_{\dot\oplus} = \{((a, b, c), (d, e, f)) \in I \times I \mid (a \cup b) \cap (d \cup e) = a \cap f = d \cap c = \emptyset\}$ with $(a, b, c) \dot\oplus (d, e, f) = (a \cup d, b \cup e, (c \cup f) \setminus (b \cup e))$.
- $dom_{\dot\|} = \{((a, b, c), (d, e, f)) \in I \times I \mid (a \cup b) \cap (d \cup e) = (a \cup b) \cap f = c \cap (d \cup e) = \emptyset\}$ with $(a, b, c) \dot\| (d, e, f) = (a \cup d, b \cup e, c \cup f)$.

Finally we have to evolve the information needed for the concurrent composition. For the concurrent occurrence rule it only has to be distinguished between places used for token flow and context places that are no flow places. This leads to the following support mapping for the definition of the greatest closed congruence:

- $supp : 2^I \to 2^P \times 2^P$, $supp\,(A) = (s_1(A), s_2(A) \setminus s_1(A))$ where $s_1(A) = \bigcup_{(a,b,c) \in A} (a \cup b)$ and $s_2(A) = \bigcup_{(a,b,c) \in A} c$.
- $\cong \subseteq 2^I \times 2^I$, $A \cong B \iff supp(A) = supp(B)$.

In [11] it was shown that $\cong$ actually is the greatest closed congruence on $\mathcal{X} = (2^I, \{\dot\|\}, dom_{\{\dot\|\}}, 2^I \times 2^I, \cup)$:

Lemma 6.1. Denote $J = \{(x, y) \in 2^P \times 2^P \mid x \cap y = \emptyset\}$. Let $\circ$ be the binary operation on J defined by $(w, c) \circ (w', c') = (w \cup w', (c \cup c') \setminus (w \cup w'))$, $dom_{\overline{\|}} = \{((w, c), (v, f)) \in J \times J \mid w \cap v = c \cap v = w \cap f = \emptyset\}$ and $\overline{\|} = \circ|_{dom_{\overline{\|}}}$. Then it holds:

(a) The mapping $supp : (2^I, \{\dot\|\}, dom_{\{\dot\|\}}, \cup) \to (J, \overline{\|}, dom_{\overline{\|}}, \circ)$ is a surjective closed homomorphism.

(b) The closed congruence $\cong \subseteq 2^I \times 2^I$ defined by $A \cong B \iff supp(A) = supp(B)$ is the greatest closed congruence on $\mathcal{X} = (2^I, \{\dot\|\}, dom_{\{\dot\|\}}, \cup)$.

Proof:
[Sketch of the proof] Part (a): Checks of the three parts of the statement: homomorphism, closeness and surjectivity. Part (b): Any congruence $\approx$ greater than $\cong$ is shown to be not closed. To show this we consider information sets $A, A' \in 2^I$ such that $A \approx A'$ but $A \not\cong A'$ and construct an information set $C \in 2^I$ fulfilling $(A, C) \in dom_{\{\dot\|\}}$ but $(A', C) \notin dom_{\{\dot\|\}}$ (by distinguishing two cases). See the Appendix for a detailed proof. □

The partial algebra $(J, \overline{\|}, dom_{\overline{\|}}, \circ)$ is isomorphic with the greatest closed congruence on $\mathcal{X}$. Therefore, by the construction of process terms (using concurrent and sequential composition) it is enough to save just the set of flow places (occurring as pre-set or post-set places of transitions in the process term) and the set of inhibiting places which are no flow places. This information suffices for deciding whether process terms are (concurrently) independent.

Moreover the following theorem from [11] reveals that the chosen implementation is correct:

Theorem 6.1. The algebraic $(\mathcal{M},\mathcal{I})$-net $(\mathcal{A}, \mathit{inf}) = ((2^P, T, pre, post), \mathit{inf})$ with $\mathcal{M}, \mathcal{I}, pre, post, \mathit{inf}$ as developed in this section corresponds to ENI (according to Definition 5.1).

Proof:
[Sketch of the proof] The two implications of the equivalence from Definition 5.1 are checked using the definition of $\mathcal{I}$. That means it is checked that the partial algebra of information $\mathcal{I}$ correctly encodes the occurrence rule for synchronous steps in ENI defined in Definition 2.3. See the Appendix for a detailed proof. □

In order to apply the general results of section 4 unrestrictedly, we check that the algebraic $(\mathcal{M},\mathcal{I})$-net implementation of this section fulfills all formulated properties of the mapping inf.

Lemma 6.2. The algebraic $(\mathcal{M},\mathcal{I})$-net $(\mathcal{A}, \mathit{inf}) = ((2^P, T, pre, post), \mathit{inf})$ with $\mathcal{M}, \mathcal{I}, pre, post, \mathit{inf}$ as developed in this section fulfills (Con1) - (Con6) and (Det).

Proof:
[Sketch of the proof] Straightforward checks. See the Appendix for a detailed proof. □

To prove the consistency of the algebraic approach with the process based concept we can use an important result about activator processes. Corollary 2 in [15] (considering the more general case of p/t-nets with inhibitor arcs) reads in our terminology:

Theorem 6.2. $\{\mathcal{S}_\alpha \mid \alpha \in \mathit{Stepseq}_{(\mathcal{A},\mathit{inf},m)}\} = \bigcup_{r \in \mathbf{Run}(ENI,m)} \mathit{strat}_{sos}(r)$.

As a consequence we directly get that $\mathbf{Run}(ENI, m) \subseteq \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. In order to prove the main result $\mathbf{Run}(ENI, m) = \mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$, we fundamentally need the following lemma.

Lemma 6.3. Let $\mathcal{S}_1 = (V, \prec_1, \sqsubset_1, id)$ and $\mathcal{S}_2 = (V, \prec_2, \sqsubset_2, id)$ be so-structures of $\sim$-equivalent process terms $\alpha : m \to m'$ and $\beta : m \to m'$ of $(\mathcal{A}, \mathit{inf})$. If $\mathcal{S} = (V, \prec, \sqsubset, id) \in \mathbf{Run}(ENI, m)$ satisfies $\mathcal{S} \subseteq \mathcal{S}_2$, then $\mathcal{S} \subseteq \mathcal{S}_1$.

Proof:
[Sketch of the proof] It is enough to consider the cases where α is derived from β through one of the equivalent transformation axioms (1)-(11) (Definition 4.3). Because for axioms preserving associated so-structures the statement is trivial we will only consider the axioms (4) and (7). We will prove the statement by contradiction. Namely, assuming that $\mathcal{S}_1$ does not extend $\mathcal{S}$ contradicts that α is a defined process term. This can in each case be computed by reducing the proof to one of the three situations shown in Figure 6. See the Appendix for a detailed argumentation. □

As a corollary we get that for each run $r \in \mathbf{Run}(ENI, m)$ there is α with $\Upsilon_r^{can} = [\alpha]_\sim$. That means $\mathbf{Run}(ENI, m) \subseteq \mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$ (Corollary 4.2). For the reverse statement observe that for $\mathcal{S} \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ every so-structure $\mathcal{S}' \in strat_{sos}(\mathcal{S})$ is associated to $\alpha \in Stepseq_{(\mathcal{A},\mathit{inf},m)}$. All these process terms α are $\sim$-equivalent (Theorem 4.2) and all elements of $strat_{sos}(\mathcal{S})$ are extensions of one run $r \in \mathbf{Run}(ENI, m)$ (Theorem 6.2, Lemma 6.3). Using the representation of $\mathcal{S}$ from Proposition 2.1, we get that $\mathcal{S}$ itself is an extension of r. This gives altogether

Theorem 6.3. Given ENI and $(\mathcal{A}, \mathit{inf})$ as defined above, there holds

$$\mathbf{Run}(ENI, m) = \mathbf{MinEnabled}(\mathcal{A}, \mathit{inf}, m)$$

.

Furthermore, we deduce that every $\sim$-equivalence class of process terms of the copy net is the canonical set of a unique run (Theorem 4.1, Remark 4.1). Consequently there holds the following one-to-one relationship, which is an enhancement of the main result of [13] (proven in another manner).

Theorem 6.4. Let ENI and $(\mathcal{A}, \mathit{inf})$ as defined above, and let the mapping $\psi : \mathbf{Run}(ENI, m) \to \{[\alpha]_\sim \mid pre(\alpha) = m\}$ be defined by $\psi(r) = [\alpha]_\sim$ for some α such that $\mathcal{S}_\alpha$ is an extension of r. Then ψ is well-defined and bijective.

Finally, this especially implies that every $\mathcal{S} \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ is an extension of exactly one $r \in \mathbf{Run}(ENI, m)$. This result is strongly connected to the well-known result obtained for elementary nets (without context), which says that each occurrence sequence of an elementary net is a linearization of exactly one run of the net.

7. Conclusion

In this paper we have presented a very flexible and general unifying approach regarding causal semantics (sketched in Figure 5). While in other approaches of unifying Petri nets (see e.g. [23, 21, 22, 14]) the occurrence rule is never a parameter. Therefore the definitions in [21] and [14] both capture elementary nets but let open more complicated restrictions of enabling conditions in the occurrence rule, such as inhibitor arcs or capacities.

We demonstrated the applicability of our schematic framework with the example of elementary nets with inhibitor arcs equipped with the a-priori semantics. The following table shows further net classes for which we developed an algebraic implementation (i.e. process terms) and references to the respective works (column "algebraic impl."). The algebraic implementation in particular includes the development of an appropriate partial algebra of information and the identification of the greatest closed congruence (the remaining parts of the instantiation of the filled arcs in Figure 5 are straightforward). The instantiation of the non-filled arcs in Figure 5 requires the existence of process semantics for the considered net class: The column "ad-hoc corr." references to works, in which a correspondence of the algebraic process term semantics and existing classical process semantics of the net class is shown in an ad-hoc way. That means a correspondence between equivalence classes of process terms and process nets is proven neglecting causal semantics. The column "causal corr." references to works, in which a correspondence of respective causal semantics is shown. An example for such an approach is section 6 in this paper. Such an approach leads to more general results than the results in the column "ad-hoc corr.".

net class	**algebraic impl.**	**ad-hoc corr.**	**causal corr.**
elementary nets	[7, 11]	[7]	[11]
elementary nets with inhibitor arcs (a-posteriori)	[8, 11]	[8]	[11]
elementary nets with read arcs (a-posteriori)	[8, 11]	[8]	[11]
elementary nets with mixed context (a-posteriori)	[8, 11]	[8]	[11]
p/t-nets	[8, 11]	–	[11]
p/t-nets with inhibitor arcs (a-posteriori)	[8, 11]	–	–
p/t-nets with weak capacities	[8, 11]	–	–
p/t-nets with strong capacities	[8, 11]	–	–
elementary nets with inhibitor arcs (a-priori)	[13, 12]	[13]	[12]
elementary nets with read arcs (a-priori)	[18]	–	–
elementary nets with mixed context (a-priori)	[18]	–	–
p/t-nets with inhibitor arcs (a-priori)	[18]	–	–
elementary nets (synchronous semantics)	[18]	–	–
p/t-nets with weak capacities (synchronous semantics)	[18]	–	–

The general algebraic part of the paper showed the extension of the basic approach from [11] to so-structure based semantics. It would be an interesting and promising project of further research to additionally extend the approach of algebraic Petri nets in order to include new net classes of a different fundamental structure or with different semantical notions. For an example besides so-structures there is another generalization of partial orders that allow interval orders as a model of system runs [26]. This would lead to another extension of the algebraic framework distinguishing the start and the end of events. But we also have more proximate and immediate research in this area. On the one hand we still have to examine some net classes and/or compare the algebraic semantics to process semantics, as for example elementary nets with read arcs and p/t-nets with inhibitor arcs each equipped with the a-priori semantics, nets with priorities or nets with reset arcs. On the other hand it would be interesting to derive behavioural results beyond the causal semantics on the abstract level. Because of the generality of the approach this could result in a very powerful analyzing tool.

References

[1] Baldan, P., Corradini, A., Montanari, U.: Contextual Petri Nets, Asymmetric Event Structures, and Processes., *Inf. Comput.*, **171**(1), 2001, 1–49.

[2] Best, E., Devillers, R.: Sequential and Concurrent Behaviour in Petri Net Theory., *Theoretical Computer Science*, **55**(1), 1987, 87–136.

[3] Bruni, R., Sassone, V.: Algebraic Models for Contextual Nets., *ICALP* (U. Montanari, J. D. P. Rolim, E. Welzl, Eds.), 1853, Springer, 2000.

[4] Burmeister, P.: *Lecture Notes on Universal Algebra – Many Sorted Partial Algebras.*, Technical report, TU Darmstadt, 2002.

[5] Busi, N., Pinna, G. M.: Process Semantics for Place/Transition Nets with Inhibitor and Read Arcs., *Fundam. Inform.*, **40**(2-3), 1999, 165–197.

[6] Desel, J., Juhás, G.: What Is a Petri Net?., in: H. Ehrig; G. Juhás; J. Padberg; G. Rozenberg [9], 1–25.

[7] Desel, J., Juhás, G., Lorenz, R.: Process Semantics of Petri Nets over Partial Algebra., in: M. Nielsen; D. Simpson [17], 146–165.

[8] Desel, J., Juhás, G., Lorenz, R.: Petri Nets over Partial Algebra., in: H. Ehrig; G. Juhás; J. Padberg; G. Rozenberg [9], 126–172.

[9] H. Ehrig; G. Juhás; J. Padberg; G. Rozenberg, Ed.: *Unifying Petri Nets, Advances in Petri Nets*, vol. 2128 of *Lecture Notes in Computer Science*, Springer, 2001.

[10] Janicki, R., Koutny, M.: Semantics of Inhibitor Nets., *Inf. Comput.*, **123**(1), 1995, 1–16.

[11] Juhas, G.: Are these Events Independend? It depends!, Habilitation, 2005.

[12] Juhas, G., Lorenz, R., Mauser, S.: Synchronous + Concurrent + Sequential = Earlier than + Not later than., *Proceedings of ACSD 2006*, 2006.

[13] Juhás, G., Lorenz, R., Singliar, T.: On Synchronicity and Concurrency in Petri Nets., *ICATPN* (W. M. P. van der Aalst; E. Best, Ed.), 2679, Springer, 2003.

[14] Kindler, E., Weber, M.: The Dimensions of Petri Nets: The Petri Net Cube., *Bulletin of the EATCS*, **66**, 1998, 155–165.

[15] Kleijn, H. C. M., Koutny, M.: Process Semantics of P/T-Nets with Inhibitor Arcs., in: M. Nielsen; D. Simpson [17], 261–281.

[16] Kleijn, H. C. M., Koutny, M.: Process semantics of general inhibitor nets., *Inf. Comput.*, **190**(1), 2004, 18–69.

[17] M. Nielsen; D. Simpson, Ed.: *Application and Theory of Petri Nets 2000, 21st International Conference, ICATPN 2000, Aarhus, Denmark, June 26-30, 2000, Proceeding*, vol. 1825 of *Lecture Notes in Computer Science*, Springer, 2000.

[18] Mauser, S.: *Sematiken von Petrinetzen - Ein algebraischer Ansatz, der zwischen nebenlufigem und synchronem Verhalten unterscheidet*, Master Thesis, Katholische Universität Eichstätt-Ingolstadt, 2006.

[19] Meseguer, J., Montanari, U.: Petri nets are monoids., *Information and Computation*, **88**(2), 1990, 105–155.

[20] Montanari, U., Rossi, F.: Contextual Nets, *Acta Inf.*, **32**(6), 1995, 545–596.

[21] Padberg, J.: *Abstract Petri Nets: Uniform Approach and Rule-Based Refinement.*, Ph.D. Thesis, TU Berlin, 1996.

[22] Padberg, J.: Classification of Petri Nets Using Adjoint Functors., *Bulletin of the EATCS*, **66**, 1998, 85–91.

[23] Padberg, J., Ehrig, H.: Parameterized Net Classes: A Uniform Approach to Petri Net Classes., in: H. Ehrig; G. Juhás; J. Padberg; G. Rozenberg [9], 173–229.

[24] Sassone, V.: The Algebraic Structure of Petri Nets., in: *Current Trends in Theoretical Computer Science*, World Scientific, 2004.

[25] Stehr, M.-O., Meseguer, J., Ölveczky, P. C.: Rewriting Logic as a Unifying Framework for Petri Nets., in: H. Ehrig; G. Juhás; J. Padberg; G. Rozenberg [9], 250–303.

[26] Vogler, W.: Partial Order Semantics and Read Arcs., *MFCS* (I. Prívara, P. Ruzicka, Eds.), 1295, Springer, 1997.

Appendix: Proofs omitted from the main text and necessary technical notions

Proof of Lemma 3.1

(i) : reflexive: $a = b \in 2^I/_{\cong}$ implies that $A \in 2^I$ exists with $a = [A] = b$. With $A \subseteq A$ follows the statement.

transitive: $a \subseteq/_{\cong} b \subseteq/_{\cong} c \Longrightarrow \exists A, B, B', C \in 2^I$ with $a = [A], b = [B] = [B'], c = [C]$ and $A \subseteq B, B' \subseteq C$. Then: $c = [C] = [C \cup B'] = [C] \cup/_{\cong} [B'] = [C] \cup/_{\cong} [B] = [C \cup B]$. With $A \subseteq C \cup B$ follows the statement.

antisymmetric: $a \subseteq/_{\cong} b$ and $b \subseteq/_{\cong} a \Longrightarrow \exists A, A', B, B' \in 2^I$ with $a = [A] = [A'], b = [B] = [B']$ and $A \subseteq B, B' \subseteq A'$. Then for arbitrary c there holds:

- $a \{\dot{\|}\}/_{\cong} c$ defined $\Longleftrightarrow b \{\dot{\|}\}/_{\cong} c$ defined.
- In the positive case: $(a \{\dot{\|}\}/_{\cong} c) \subseteq/_{\cong} (b \{\dot{\|}\}/_{\cong} c)$ and $(b \{\dot{\|}\}/_{\cong} c) \subseteq/_{\cong} (a \{\dot{\|}\}/_{\cong} c)$.
- $(a \cup/_{\cong} c) \subseteq/_{\cong} (b \cup/_{\cong} c)$ and $(b \cup/_{\cong} c) \subseteq/_{\cong} (a \cup/_{\cong} c)$.

Because $\cong$ is a closed congruence, one can observe, that the properties of a closed congruence are maintained, if all equivalence classes a, b with $a \subseteq/_{\cong} b$ and $b \subseteq/_{\cong} a$ are identified in one equivalence class. Since $\cong$ is the greatest closed congruence, there already holds $a = b$.

(ii) + (iii) : The preliminaries $a \subseteq/_{\cong} a'$, $b \subseteq/_{\cong} b'$ can be rewritten as follows: $\exists A, A', B, B' \in 2^I$ with $a = [A], a' = [A'], b = [B], b' = [B']$ and $A \subseteq A', B \subseteq B'$. We start with the first part of (iii): $a \cup/_{\cong} b = [A] \cup/_{\cong} [B] = [A \cup B]$ and $a' \cup/_{\cong} b' = [A'] \cup/_{\cong} [B'] = [A' \cup B']$. With $(A \cup B) \subseteq (A' \cup B')$ follows $(a \cup/_{\cong} b) \subseteq/_{\cong} (a' \cup/_{\cong} b')$. Now we additionally assume that $a' \{\dot{\|}\}/_{\cong} b'$ is defined, i.e. $A'\{\dot{\|}\}B'$ is defined. According to the definition of $\{\dot{\|}\}$ there results: $A\{\dot{\|}\}B$ defined with $A\{\dot{\|}\}B \subseteq A'\{\dot{\|}\}B'$. Consequently $a \{\dot{\|}\}/_{\cong} b$ is defined and we can calculate: $a \{\dot{\|}\}/_{\cong} b = [A] \{\dot{\|}\}/_{\cong} [B] = [A\{\dot{\|}\}B]$ and $a' \{\dot{\|}\}/_{\cong} b' = [A'] \{\dot{\|}\}/_{\cong} [B'] = [A'\{\dot{\|}\}B']$, consequently $(a \{\dot{\|}\}/_{\cong} b) \subseteq/_{\cong} (a' \{\dot{\|}\}/_{\cong} b')$. Therefore (ii) and the second statement of (iii) are proven.

(iv) : Let $A, B \in 2^I$ with $a = [A], b = [B]$. There holds $a \cup/_{\cong} b = [A] \cup/_{\cong} [B] = [A \cup B]$, which leads to the first statement. Let now $A, B \in 2^I$, whereas for any information element $m \in A$ there holds: $\exists i \in I : m \,\dot{\|}\, i \in B$. Defining $\cong'$ in such a manner, that for any such sets A, B the relation $B \cong' (A \cup B)$ holds and otherwise $\cong'$ coincides with $\cong$, then $\cong'$ is again a closed congruence (simple observation using the definition of $\{\dot{\|}\}$). Since $\cong$ is the greatest closed congruence, this relation already holds for $\cong$, i.e. $\cong = \cong'$. Consequently there are sets $A, B \in 2^I$ with $a = [A]$ and $(a \{\dot{\|}\}/_{\cong} b) = [B]$, whereupon A and B have the above relation. Therefore the second statement follows with $[B] = [A \cup B]$.

(v) : The preliminaries imply: $\exists A, B, C, C' \in 2^I$ with $a = [A], b = [B], c = [C] = [C']$ and $A \subseteq C, B \subseteq C'$. Similar to the latter argumentation we can verify with the properties of the greatest closed congruence that the relation $c = [C \cup C']$ holds (use $c = [C] = [C']$) which leads to the statement.

Proof of Lemma 4.1

We prove the statement inductively according to the construction of process terms:

If $\alpha = m \in M$, then $C \subseteq B_\alpha = \emptyset$, $\alpha_{su} = m$ and $m_C = m$, i.e. $\alpha_C = m; m$ is a defined process term obviously fulfilling (I) (axiom (8)) and (II).

If $\alpha \in Step_{(\mathcal{A},inf)}$ and $C = \emptyset$, then $\alpha_{su} = \alpha$ and $m_C = pre(\alpha)$, i.e. $\alpha_C = pre(\alpha); \alpha$ is a defined process term obviously fulfilling (I) (axiom (8)) and (II) (use (Con3)).

If $\alpha \in Step_{(\mathcal{A},inf)}$ and $C = \{\alpha\}$, then $\alpha_{su} = post(\alpha)$ and $m_C = \underline{0}$, i.e. $\alpha_C = (\alpha \parallel \underline{0}); post(\alpha)$ is a defined process term (use (Con2)), obviously fulfilling (I) ((11) and (8)) and (II) (use (Con2) and (Con3)).

Let $\alpha = \beta; \gamma$ for process terms β and γ which fulfill the statement. According to Definition 4.1 there exist disjoint sets $V_\beta \subseteq V$ and $V_\gamma \subseteq V$ with $V_\beta \cup V_\gamma = V$, such that $\mathcal{S}_\beta = (V_\beta, \prec \cap (V_\beta \times V_\beta), \sqsubset \cap (V_\beta \times V_\beta), id)$ is a labelled so-structure of β, $\mathcal{S}_\gamma = (V_\gamma, \prec \cap (V_\gamma \times V_\gamma), \sqsubset \cap (V_\gamma \times V_\gamma), id)$ is a labelled so-structure of γ and $v \prec v'$ for each $v \in V_\beta$, $v' \in V_\gamma$. B_β and B_γ are disjoint sets satisfying $B_\beta \cup B_\gamma = B_\alpha$ and consequently $X|_{\mathcal{S}_\beta} \subseteq X|_{\mathcal{S}_\alpha}$ and $X|_{\mathcal{S}_\gamma} \subseteq X|_{\mathcal{S}_\alpha}$ are disjoint sets fulfilling $X|_{\mathcal{S}_\beta} \cup X|_{\mathcal{S}_\gamma} = X|_{\mathcal{S}_\alpha}$. Moreover, $k <_{\mathcal{S}_\alpha} k'$ for each $k \in X|_{\mathcal{S}_\beta}$ and $k' \in X|_{\mathcal{S}_\gamma}$. Because α has no sub-term of the form $m; \alpha'$ we have $V_\beta \neq \emptyset$ and thus $B_\beta \neq \emptyset$ and $X|_{\mathcal{S}_\beta} \neq \emptyset$. It easily follows $C \subseteq B_\beta$ (elements of B_γ are not minimal in α). Obviously the elements of C are also minimal in β. Because also β has clearly no sub-term of the form $m; \alpha'$ and β fulfills the statement, there exists a marking m_C, such that $\beta_C = ((\|_{s \in C}\, s) \parallel m_C); \beta_{su}$ is a defined process term fulfilling (I) and (II). Furthermore we have $\gamma_{su} = \gamma$, because B_γ and C are disjoint. Consequently $\alpha_C = ((\|_{s \in C}\, s) \parallel m_C); \alpha_{su} = ((\|_{s \in C}\, s) \parallel m_C); \beta_{su}; \gamma_{su} = \beta_C; \gamma$ is a defined process term (because $post(\beta_C) = post(\beta) = pre(\gamma)$). From $\beta_C \sim \beta$ we get (I) and from $Inf(\beta_C) \subseteq/_{\cong} Inf(\beta)$ we can conclude (II).

Let $\alpha = \beta \parallel \gamma$ for process terms β and γ which fulfill the statement. As in the previous paragraph there exist disjoint sets $V_\beta \subseteq V$ and $V_\gamma \subseteq V$ with $V_\beta \cup V_\gamma = V$, such that $\mathcal{S}_\beta = (V_\beta, \prec \cap (V_\beta \times V_\beta), \sqsubset \cap (V_\beta \times V_\beta), id)$ is a labelled so-structure of β, $\mathcal{S}_\gamma = (V_\gamma, \prec \cap (V_\gamma \times V_\gamma), \sqsubset \cap (V_\gamma \times V_\gamma), id)$ is a labelled so-structure of γ and there holds $v \not\sqsubset v'$ as well as $v' \not\sqsubset v$ for each $v \in V_\beta$, $v' \in V_\gamma$. Again B_β and B_γ are disjoint sets satisfying $B_\beta \cup B_\gamma = B_\alpha$, $X|_{\mathcal{S}_\beta} \subseteq X|_{\mathcal{S}_\alpha}$ and $X|_{\mathcal{S}_\gamma} \subseteq X|_{\mathcal{S}_\alpha}$ are disjoint sets fulfilling $X|_{\mathcal{S}_\beta} \cup X|_{\mathcal{S}_\gamma} = X|_{\mathcal{S}_\alpha}$ and $k \not<_{\mathcal{S}_\alpha} k'$ and $k' \not<_{\mathcal{S}_\alpha} k$ for each $k \in X|_{\mathcal{S}_\beta}$, $k' \in X|_{\mathcal{S}_\gamma}$. Define $C_1 = C \cap B_\beta$ and $C_2 = C \cap B_\gamma$. Then C_1 and C_2 are disjoint with $C_1 \cup C_2 = C$. Obviously C_1 is a subset of minimal basic step terms of β and C_2 is a subset of minimal basic step terms of γ. Because β and γ have no sub-term of the form $m; \alpha'$, there exist markings m_1 and m_2, such that $\beta_{C_1} = ((\|_{s \in C_1}\, s) \parallel m_1); \beta_{su}$ and $\gamma_{C_2} = ((\|_{s \in C_2}\, s) \parallel m_2); \gamma_{su}$ are defined process terms fulfilling (I) and (II). From $Inf(\beta_{C_1}) \subseteq/_{\cong} Inf(\beta)$ and $Inf(\gamma_{C_2}) \subseteq/_{\cong} Inf(\gamma)$ we derive that $\beta_{C_1} \parallel \gamma_{C_2}$ is defined with $Inf(\beta_{C_1} \parallel \gamma_{C_2}) \subseteq/_{\cong} Inf(\beta \parallel \gamma) = Inf(\alpha)$. For $m_C = m_1 + m_2$ we get the following relation using axiom (4) in the first equivalent transformation step and the axioms (1), (2) and (9) in the second transformation step: $\beta_{C_1} \parallel \gamma_{C_2} = (((\|_{s \in C_1}\, s) \parallel m_1); \beta_{su}) \parallel (((\|_{s \in C_2}\, s) \parallel m_2); \gamma_{su}) \sim (((\|_{s \in C_1}\, s) \parallel m_1) \parallel ((\|_{s \in C_2}\, s) \parallel m_2)); (\beta_{su} \parallel \gamma_{su}) \sim ((\|_{s \in C}\, s) \parallel m_C); \alpha_{su} = \alpha_C$. Consequently α_C is a defined process term. With (I) for β_{C_1} and γ_{C_2} we get (I) for α_C. From the used $\sim$-axioms and (II) for β_{C_1} and γ_{C_2} there results $Inf(\alpha_C) \subseteq/_{\cong} Inf(\beta_{C_1} \parallel \gamma_{C_2}) \subseteq/_{\cong} Inf(\beta \parallel \gamma) = Inf(\alpha)$. Consequently (II) is fulfilled.

Definition of copy terms

In order to formulate basic relations between process terms of a copy-net $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ and of $(\mathcal{A}, \mathit{inf})$, we extend the labelling function l from Definition 4.4 to process terms α of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ in the following way for markings m and process terms α_1, α_2: $l(m) = m$, $l(\alpha_1 \oplus \alpha_2) = l(\alpha_1) \oplus l(\alpha_2)$, $l(\alpha_1 \parallel \alpha_2) = l(\alpha_1) \parallel l(\alpha_2)$ and $l(\alpha_1; \alpha_2) = l(\alpha_1); l(\alpha_2)$ (if defined, respectively). The extended labelling function l fulfills the following immediate properties for process terms α, β of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ and α' of $(\mathcal{A}, \mathit{inf})$:

- $l(\alpha) \in \mathcal{P}(\mathcal{A}, \mathit{inf})$.
- $\mathit{pre}(\alpha) = \mathit{pre}(l(\alpha))$, $\mathit{post}(\alpha) = \mathit{post}(l(\alpha))$ and $\mathit{Inf}(\alpha) = \mathit{Inf}(l(\alpha))$.
- $\alpha \sim \beta \Longrightarrow l(\alpha) \sim l(\beta)$.
- $l : \mathcal{P}(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)}) \to \mathcal{P}(\mathcal{A}, \mathit{inf})$ is surjective.
- $\mathcal{S} = (V, \prec, \sqsubset, id)$ associated to $\alpha \Longrightarrow \mathcal{S}' = (V, \prec, \sqsubset, l)$ associated to $l(\alpha)$.
- $\mathcal{S}' = (V, \prec, \sqsubset, l)$ associated to $\alpha' \Longrightarrow \exists \alpha$ with $l(\alpha) = \alpha'$ such that $\mathcal{S} = (V, \prec, \sqsubset, id)$ is associated to α.
- $\mathcal{S} = (V, \prec, \sqsubset, id)$ and $\mathcal{S}' = (V, \prec', \sqsubset', id)$ associated to $\alpha \Longrightarrow \mathcal{S} = \mathcal{S}'$.

Now we are prepared to define the technically important copy terms: Let $\mathcal{S} = (V, \prec, \sqsubset, l)$ be an so-structure of a process term α of $(\mathcal{A}, \mathit{inf})$. A *copy term of α (w.r.t. $\mathcal{S}$)* is a process term $\alpha^{\mathcal{S}}$ of the (V, l)-copy net $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ with $l(\alpha^{\mathcal{S}}) = \alpha$ such that $(V, \prec, \sqsubset, id)$ is associated to $\alpha^{\mathcal{S}}$.

Two copy terms of α w.r.t. $\mathcal{S}$ are always $\sim$-equivalent through commutativity and associativity axioms. We will not further distinguish such copy terms.

Proof of Theorem 4.1

Denote $\mathcal{S} = \mathcal{S}_\alpha = (V, \prec, \sqsubset, l)$ and $\mathcal{S}' = (V, \prec', \sqsubset', l)$. We will equivalently transform $\alpha^{\mathcal{S}}$ into the process term $\beta^{\mathcal{S}'}$, i.e. $\beta^{\mathcal{S}'}$ has associated the so-structure $(V, \prec', \sqsubset', id)$, then the process term $\beta = l(\beta^{\mathcal{S}'})$ has associated the so-structure $\mathcal{S}'$ and satisfies $\alpha \sim \beta$. Consequently $\mathit{pre}(\beta) = \mathit{pre}(\alpha) = m$ and thus $\mathcal{S}$ is enabled in m.

Let $po_{\mathcal{S}} = (X|_{\mathcal{S}}, <_{\mathcal{S}})$ be the partial order associated to $\mathcal{S}$, let $po_{\mathcal{S}'} = (X|_{\mathcal{S}'}, <_{\mathcal{S}'})$ be the total order associated to $\mathcal{S}'$ (see Lemma 2.2). Let $X|_{\mathcal{S}'} = \{k'_1, \ldots, k'_m\}$ such that $k'_i <_{\mathcal{S}'} k'_j \iff i < j$. Then k'_1 is of the form $k'_1 = k_1 \cup \ldots \cup k_n$ for (pairwise disjoint) $k_1, \ldots, k_n \in X|_{\mathcal{S}}$, where $k_1, \ldots, k_n$ are all minimal w.r.t. $<_{\mathcal{S}}$ (see Lemma 2.2). Thus, the basic step terms $b_1, \ldots, b_n \in B_{\alpha^{\mathcal{S}}}$ of $\alpha^{\mathcal{S}}$ corresponding to $k_1, \ldots, k_n$ are minimal.

Define $C = \{b_1, b_2, \ldots, b_n\}$ and $su = post|_C$. Without loss of generality we can suppose that $\alpha^{\mathcal{S}}$ has no sub-term of the form $m; \alpha'$. According to corollary 4.1 there exists a marking m_{Sy} such that $\alpha^{\mathcal{S}}_C = ((\oplus_{s \in C}\, s) \parallel m_{Sy}); \alpha^{\mathcal{S}}_{su} = ((\oplus_{v \in k'_1}\, v) \parallel m_{Sy}); \alpha^{\mathcal{S}}_{su}$ is a defined process term fulfilling (I) and (II). Thus we have detached k'_1 from $\alpha^{\mathcal{S}}$. Denote $V' = V \setminus \{v \mid \exists s \in C \text{ with } v \in s\}$, $K_C = X|_{\mathcal{S}} \setminus \{k_1, \ldots, k_n\}$ and $K'_C = X|_{\mathcal{S}'} \setminus \{k'_1\}$. Directly from the construction of $\alpha^{\mathcal{S}}_{su}$ we get that $\mathcal{S}_C = (V', \prec|_{V' \times V'}, \sqsubset|_{V' \times V'}, id)$ is an so-structure of $\alpha^{\mathcal{S}}_{su}$ and $po_{\mathcal{S}_C} = (K_C, <_{\mathcal{S}}|_{K_C \times K_C})$ is the associated partial order.

Observe now that $\mathcal{S}'_C = (V', \prec' |_{V'\times V'}, \sqsubset' |_{V'\times V'}, id)$ is a total linear extension of $\mathcal{S}_C$, $k'_2 \in X|_{\mathcal{S}'}$ is minimal in $po_{\mathcal{S}'_C} = (K'_C, <_{\mathcal{S}'} |_{K'_C\times K'_C})$. That means we can re-iterate the above procedure for k'_2, $\alpha^{\mathcal{S}}_{su}$, $\mathcal{S}_C$ and $\mathcal{S}'_C$ instead of k'_1, α, $\mathcal{S}$ and $\mathcal{S}'$, and subsequently for all further $k'_3, \ldots, k'_m \in X|_{\mathcal{S}'}$. This results in the searched process term $\beta' = ((\oplus_{v\in k'_1} v) \parallel m_1); \ldots ; ((\oplus_{v\in k'_m} v) \parallel m_k)(= \beta^{\mathcal{S}'})$.

Determinism properties of $\sim$

We can conclude the following technical results (Det1) and (Det2) concerning the determinism of markings.

(Det1) If $(\mathcal{A}, \mathit{inf})$ fulfills (Det), then there holds for $s \in Step_{(\mathcal{A},\mathit{inf})}$ and $m_1, m_2 \in M$: $(s \parallel m_1$ defined $\wedge\, s \parallel m_2$ defined $\wedge\, pre(s \parallel m_1) = pre(s \parallel m_2)) \Longrightarrow (post(s \parallel m_1) = post(s \parallel m_2) \wedge (s \parallel m_1) \sim (s \parallel m_2))$.

(Det2) If $(\mathcal{A}, \mathit{inf})$ fulfills (Det), (Con1) and (Con3)-(Con5), then there holds for $s_1, s_2 \in Step_{(\mathcal{A},\mathit{inf})}$ and $m, m_1, m_2 \in M$: $((s_1 \oplus s_2) \parallel m$ defined $\wedge\, (s_1 \parallel m_1); (s_2 \parallel m_2)$ defined $\wedge\, pre((s_1 \oplus s_2) \parallel m) = pre((s_1 \parallel m_1); (s_2 \parallel m_2))) \Longrightarrow (post((s_1 \oplus s_2) \parallel m) = post((s_1 \parallel m_1); (s_2 \parallel m_2)) \wedge (s_1 \oplus s_2) \parallel m \sim (s_1 \parallel m_1); (s_2 \parallel m_2))$.

The two statements can be proven as follows:

(Det1): The first implication concerning $post$ directly results from (Det); the equivalence of the two terms then yields from (10).

(Det2): The preliminaries exactly coincide with the preliminaries in the definition of (Con5). Therefore $s_1 \parallel (pre(s_2) + m)$ and $s_2 \parallel (post(s_1) + m)$ are defined and thus also $(s_1 \parallel (pre(s_2) + m)); (s_2 \parallel (post(s_1) + m))$ is defined. The initial marking of the first term $s_1 \parallel (pre(s_2) + m)$ coincides with the initial marking of $s_1 \parallel m_1$. Consequently (Det) implies, that the final markings are equal and from (10) results $s_1 \parallel (pre(s_2) + m) \sim s_1 \parallel m_1$. Because the final markings of these two terms coincide, the initial markings of the next two terms $s_2 \parallel (post(s_1) + m)$ and $s_2 \parallel m_2$ have to be equal. With the same arguments there also results the coincidence of the final markings and the $\sim$-equivalence. Note that with (Con4) and (Con3) $pre(s_2) \parallel m$ and $post(s_1) \parallel m$ are defined. The following equivalence transformation finally yields the statement: $(s_1 \parallel (pre(s_2) + m)); (s_2 \parallel (post(s_1) + m)) \sim (s_1 \parallel pre(s_2) \parallel m); (s_2 \parallel post(s_1) \parallel m) \sim ((s_1 \parallel pre(s_2)); (s_2 \parallel post(s_1))) \parallel m \sim (s_1 \oplus s_2) \parallel m$. Thereby firstly (9), then (4) and (8) and lastly (7) was applied.

(Det1) shows that $s \parallel m_1$ is equivalent to $s \parallel m_2$ if the initial markings of the two process terms coincide. In particular we do not have to require $m_1 = m_2$. That means in this situation we do not have to assume a uniqueness property for concurrently composed markings in order to get equivalent process terms. (Det2) similarly shows that we do not have to make uniqueness assumptions on the concurrently composed markings m_1, m_2, m when equivalently transforming a synchronous step term $s_1 \oplus s_2$ into a sequence of the sub terms s_1 and s_2. In particular we do not have to require $m_2 = m + post(s_1)$ or $m_1 = m + pre(s_2)$. Again we only have to assume that the initial markings of the process terms coincide.

Proof of Lemma 4.2

Denote $\mathcal{S} = (V, \prec, \sqsubset, l)$, $\mathcal{S}' = (V, \prec', \sqsubset', l)$ and $\mathcal{S}'' = (V, \prec'', \sqsubset'', l)$. We show that both α and β can be equivalently transformed to equivalent synchronous step sequence terms γ_α resp. γ_β which only depend on $\mathcal{S}$. As usual this is done on the copy term level. We start with the process term α. Without loss of generality $\alpha^{\mathcal{S}'}$ is of the form $\alpha^{\mathcal{S}'} = (s_1 \parallel m_1); \ldots ; (s_n \parallel m_n)$ with $s_i \in Step_{(\mathcal{A}_{(V,l)}, inf_{(V,l)})}$ and $m_i \in M$ for $i = 1, \ldots, n$. Let $V_{min} \subseteq V$ be the set of minimal events in $\mathcal{S}$ w.r.t. $\prec$. Without loss of generality each step term s_i is of the form $s_i = (s_i^{min} \oplus s_i^{rest})$, where $s_i^{min}, s_i^{rest} \in Step_{(\mathcal{A}_{(V,l)}, inf_{(V,l)})}$ with $V_{min} \cap |s_i| = |s_i^{min}|$. Of course the terms s_i^{min} and s_i^{rest} are also allowed to be empty. In the following equivalence transformations s_i^{min} and s_i^{rest} are always considered not to be empty, because in the case that one term is empty there is always an obvious way to justify the given transformation.

First we equivalently sequentialize each $s_i \parallel m_i$ into $(s_i^{min} \parallel m_i^{min}); (s_i^{rest} \parallel m_i^{rest})$. Obviously for $v \in s_i^{min}, w \in s_i^{rest}$ we have $v \sqsubset' w$ and $w \sqsubset' v$. Because w is not minimal in $\mathcal{S}$ and v is minimal in $\mathcal{S}$ we conclude $w \not\sqsubset v$ (use (C4)). Consequently if we remove for each $i \in \{1, \ldots, n\}$ and each $v \in s_i^{min}, w \in s_i^{rest}$ the relation $w \sqsubset' v$ from $\mathcal{S}'$ and add in exchange the relation $v \prec' w$ to $\mathcal{S}'$ the result is a total linear so-structure $\mathcal{S}_1' = (V, \prec_1', \sqsubset_1', l)$ which extends $\mathcal{S}$. Since $\mathcal{S}$ is enabled, there exists a synchronous step sequence term α_1' with $pre(\alpha_1') = m$ and associated so-structure $\mathcal{S}_1'$. Its copy term $(\alpha_1')^{\mathcal{S}_1'}$ has without loss of generality the form $(\alpha_1')^{\mathcal{S}_1'} = (s_1^{min} \parallel m_1^{min}); (s_1^{rest} \parallel m_1^{rest}) ; \ldots ; (s_n^{min} \parallel m_n^{min}) ; (s_n^{rest} \parallel m_n^{rest})$ satisfying $(\alpha_1')^{\mathcal{S}_1'} \sim \alpha^{\mathcal{S}'}$ (use (Det2) iteratively).

Next we iteratively equivalently permute synchronous step terms with minimal events ("min"-terms) and synchronous step terms with not minimal events ("rest"-terms), starting from behind. Analogously as above we conclude for each $v \in s_n^{min}, w \in s_{n-1}^{rest}$ that $w \not\sqsubset v$. Consequently if we remove for each $v \in s_n^{min}, w \in s_{n-1}^{rest}$ the relation $w \prec_1' v$ from $\mathcal{S}_1'$ and add in exchange the relation $v \sqsubset_1' w$ to $\mathcal{S}_1'$ the result is a total linear so-structure $\mathcal{S}_2' = (V, \prec_2', \sqsubset_2', l)$ which extends $\mathcal{S}$. As above there is a synchronous step sequence term α_2' with associated so-structure $\mathcal{S}_2'$ and the copy term of α_2' w.r.t. $\mathcal{S}_2'$ has without loss of generality the form $(s_1^{min} \parallel m_1^{min_1}); (s_1^{rest} \parallel m_1^{rest_1}) ; \ldots ; ((s_n^{min} \oplus s_{n-1}^{rest}) \parallel m_{n-1}^{minrest_1}); (s_n^{rest} \parallel m_n^{rest_1})$ being equivalent to $(\alpha_1')^{\mathcal{S}_1'}$ (using (Det1) and (Det2)) and consequently to $\alpha^{\mathcal{S}'}$. From $\mathcal{S}_2'$ we now remove each relation $w \sqsubset_2' v$ and add in exchange $v \prec_2' w$ for $v \in s_n^{min}, w \in s_{n-1}^{rest}$ resulting in the total linear so-structure $\mathcal{S}_3' = (V, \prec_3', \sqsubset_3', l)$ which also extends $\mathcal{S}$. With the same arguments we get an accordant copy term of the form $(s_1^{min} \parallel m_1^{min_2}); (s_1^{rest} \parallel m_1^{rest_2}); \ldots ; (s_n^{min} \parallel m_n^{min_2}) ; (s_{n-1}^{rest} \parallel m_{n-1}^{rest_2}); (s_n^{rest} \parallel m_n^{rest_2})$ equivalent to $\alpha^{\mathcal{S}'}$. Altogether the terms s_n^{min} and s_{n-1}^{rest} have been permuted. With a similar argumentation we can equivalently transform the sub-term $(s_{n-1}^{min} \parallel m_{n-1}^{min_2});(s_n^{min} \parallel m_n^{min_2})$ into $((s_{n-1}^{min} \oplus s_n^{min}) \parallel m^{min_3})$.

Repeating this procedure, "min"-terms are iteratively permutated with "rest"-terms from the right to the left and synchronously composed with other "min"-terms to one collective "min"-term. The result is a synchronous step sequence term of the form $(s_1^{min} \oplus \ldots \oplus s_n^{min} \parallel m^{min'}); (s_1^{rest} \parallel m_1^{rest'}) ; \ldots ; (s_{n-1}^{rest} \parallel m_{n-1}^{rest'}) ; (s_n^{rest} \parallel m_n^{rest'}) = ((\oplus_{v \in V_{min}} v) \parallel m^{min'}) ; (s_1^{rest} \parallel m_1^{rest'}) ; \ldots ; (s_{n-1}^{rest} \parallel m_{n-1}^{rest'}) ; (s_n^{rest} \parallel m_n^{rest'})$ equivalent to $\alpha^{\mathcal{S}'}$ and with an associated so-structure $\mathcal{S}'''$ extending $\mathcal{S}$. Thus, we have sorted the minimal events V_{min} of $\mathcal{S}$ to one synchronous step at the beginning of the term. Considering the so-structure $\mathcal{S}_1 = (V \setminus V_{min}, \prec|_{(V \setminus V_{min}) \times (V \setminus V_{min})}, \sqsubset|_{(V \setminus V_{min}) \times (V \setminus V_{min})}, l)$ restricting $\mathcal{S}$ to the set of remaining events, we can collect in the same way the minimal events of $\mathcal{S}_1$ to one synchronous step term at the second position of the synchronous step sequence term[11]. Now we re-iterate this procedure

[11] One must observe that the enabled property of $\mathcal{S}$ (and not of $\mathcal{S}_1$) has to be used.

until $\mathcal{S}_i$ constructed in this way is empty. Because every $\mathcal{S}_i$ has minimal events the procedure terminates. Altogether we get a synchronous step sequence term $\alpha^{\mathcal{S}'}_{\mathcal{S}_{nat}}$ equivalent to $\alpha^{\mathcal{S}'}$ that is uniquely defined by $\mathcal{S}$ up to concurrently composed markings and commutativity and associativity axioms[12]. The same can be applied to the process term β resulting in a copy term $\beta^{\mathcal{S}''}_{\mathcal{S}_{nat}}$ equivalent to $\beta^{\mathcal{S}''}$. We deduce $\alpha^{\mathcal{S}'}_{\mathcal{S}_{nat}} \sim \beta^{\mathcal{S}''}_{\mathcal{S}_{nat}}$ from $pre(\beta^{\mathcal{S}''}) = pre(\alpha^{\mathcal{S}'})$, $\alpha^{\mathcal{S}'}_{\mathcal{S}_{nat}} \sim \alpha^{\mathcal{S}'}$ and $\beta^{\mathcal{S}''}_{\mathcal{S}_{nat}} \sim \beta^{\mathcal{S}''}$ (use (Det1)), so there results $\alpha^{\mathcal{S}'} \sim \beta^{\mathcal{S}''}$ and consequently $\alpha \sim \beta$.

Proof of Lemma 6.1

The operation $\circ$ is well-defined because for any $x, y \in J$, we have $x \circ y \in J$ by construction.

ad (a): First we show that $supp$ is a homomorphism for the operations $\{\dot{\|}\}$ and $\cup$ on 2^I, that is

$$\begin{aligned} supp\,(A\{\dot{\|}\}A') &= supp\,(A)\overline{\|}supp\,(A') \\ supp\,(A \cup A') &= supp\,(A) \circ supp\,(A'). \end{aligned}$$

We compute $supp\,(A\,\{\dot{\|}\}\,A') = supp\,(\{(a \cup d, b \cup e, c \cup f) \mid (a,b,c) \in A,\ (d,e,f) \in A'\}) = (\bigcup_{(a,b,c)\in A\cup A'}(a \cup b), \bigcup_{(a,b,c)\in A\cup A'} c \setminus \bigcup_{(a,b,c)\in A\cup A'}(a \cup b)) = (\bigcup_{(a,b,c)\in A}(a \cup b) \cup \bigcup_{(a,b,c)\in A'}(a \cup b), (\bigcup_{(a,b,c)\in A} c \cup \bigcup_{(a,b,c)\in A'} c) \setminus (\bigcup_{(a,b,c)\in A}(a \cup b) \cup \bigcup_{(a,b,c)\in A'}(a \cup b))) = supp\,(A)\overline{\|}supp\,(A')$ (whenever both sides are defined). The equation $supp\,(A \cup A') = supp\,(A) \circ supp\,(A')$ follows directly from the definitions.

Next we show the closedness of $supp$, that is

$$(A, A') \in dom_{\{\dot{\|}\}} \Longleftrightarrow (supp\,(A), supp\,(A')) \in dom_{\overline{\|}}$$

for any two $A, A' \subseteq I$. Denote $s_1(A) = \overline{w}$, $s_2(A) \setminus s_1(A) = \overline{c}$, $s_1(A') = \overline{w}'$ and $s_2(A') \setminus s_1(A') = \overline{c}'$. Then it holds: $(\forall (a,b,c) \in A,\ \forall (a',b',c') \in A' :\ (a \cup b) \cap (a' \cup b') = (a \cup b) \cap c' = (a' \cup b') \cap c = \emptyset) \Longleftrightarrow (\overline{w} \cap \overline{w}' = \overline{w} \cap \overline{c}' = \overline{w}' \cap \overline{c} = \emptyset)$.

Finally, the mapping $supp$ is surjective, because, for any $(w, c) \in J$, we have $supp\,(\{(w, \emptyset, c)\}) = (w, c)$.

ad (b): We will show that any congruence $\approx$ such that $\cong$ is a proper subset of $\approx$ is not closed. Assume there are $A, A' \in 2^I$ such that $A \approx A'$ but $A \ncong A'$. Then $supp(A) \neq supp(A')$.

We define a set $C \in 2^I$ such that $(A, C) \in dom_{\{\dot{\|}\}}$ but $(A', C) \notin dom_{\{\dot{\|}\}}$ or vice versa (which implies that $\approx$ is not closed). If $supp(A) = (\overline{w}, \overline{c})$ and $supp(A') = (\overline{w}', \overline{c}')$, then $\overline{w} \cap \overline{c} = \overline{w}' \cap \overline{c}' = \emptyset$ (by definition) and $\overline{c} \neq \overline{c}' \vee \overline{w} \neq \overline{w}'$ (since $supp(A) \neq supp(A')$).

Let $\overline{w} \neq \overline{w}'$. Without loss of generality we assume $\overline{w}' \setminus \overline{w} \neq \emptyset$. Set $C = \{(\emptyset, \emptyset, c)\}$ with $c = \overline{w}' \setminus \overline{w}$. From $c \cap \overline{w} = \emptyset$, but $c \cap \overline{w}' \neq \emptyset$ we conclude $(A, C) \in dom_{\{\dot{\|}\}}$, but $(A', C) \notin dom_{\{\dot{\|}\}}$.

Now let $\overline{w} = \overline{w}'$ and $\overline{c} \neq \overline{c}'$. Without loss of generality we assume $\overline{c}' \setminus \overline{c} \neq \emptyset$. Set $C = \{(a, \emptyset, \emptyset)\}$ with $a = (\overline{c}' \setminus \overline{c})$. From $a \cap \overline{w} = a \cap \overline{c} = \emptyset$, but $a \cap \overline{c}' \neq \emptyset$ we again conclude $(A, C) \in dom_{\{\dot{\|}\}}$, but $(A', C) \notin dom_{\{\dot{\|}\}}$.

[12]This can directly be derived from the construction rule.

Proof of Theorem 6.1

Let $\tilde{s} \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$. First we show (by contradiction) that $|s|$ is a set, i.e. for each $t \in T$ we have if $|\tilde{s}|(t) > 0$ then $|\tilde{s}|(t) = 1$. Suppose that there is a $t \in T$ which appears more than once in $\tilde{s}$. Then (independent from the construction order of $\tilde{s}$) the construction of $\tilde{s}$ involves a sub-term $s' \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ of $\tilde{s}$ and $s_1, s_2 \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ such that $s' = s_1 \oplus s_2$ and t is a sub-term of s_1 as well as a sub-term of s_2. Notice that we suppose nets with transitions having nonempty pre-set or post-set, i.e. $\mathit{pre}(t) \cup \mathit{post}(t) \neq \emptyset$. Denoting $\mathit{inf}(s_1) = (a,b,c)$ and $\mathit{inf}(s_2) = (d,e,f)$, we can conclude from the definition of inf that $\mathit{pre}(t) \subseteq a \cap d$ and $\mathit{post}(t) \subseteq b \cap e$. Therefore from the definition of $\mathit{dom}_{\dot{\oplus}}$ the process term s' is not defined.

Now we show that for every step $s \in 2^T$ enabled to occur in ENI with $m \xrightarrow{s} m'$ ($m, m' \in M$) there exists $\tilde{s} \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ with $|\tilde{s}| = s$. We prove this statement by induction on the number of elements in the set s. If $s = \{t\}$ ($t \in T$), then obviously $\tilde{s} = t \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$. Let $s = \{t_1, \ldots t_k\}$ with $k > 1$ and $s^{sub} = s \setminus \{t_k\}$. Take the term s' with $|s'| = s^{sub}$. Then $\mathit{inf}(s') = (a,b,c) = (\cup_{t \in s'} \mathit{pre}(t), \cup_{t \in s'} \mathit{post}(t), (\cup_{t \in s'} {}^{-}t) \setminus b)$. Now, we can easily show (by contradiction) that $\tilde{s}$ exists: If not, then $(\mathit{inf}(s'), \mathit{inf}(t_k)) \notin \mathit{dom}_{\dot{\oplus}}$, i.e. $(a \cup b) \cap (\mathit{pre}(t_k) \cup \mathit{post}(t_k)) \neq \emptyset$ or $a \cap {}^{-}t_k \neq \emptyset$ or $c \cap \mathit{pre}(t_k) \neq \emptyset$. In each case t_k is in synchronous conflict to some $t \in s'$. Thus $t \in s^{sub}$ and t_k cannot be in the enabled step s. A contradiction.

A transition $t \in T$ enabled to occur in m fulfills $(m \setminus \mathit{pre}(t)) \cap \mathit{post}(t) = \emptyset$. This implies $(m \setminus \mathit{pre}(\tilde{s})) \cap \mathit{post}(\tilde{s}) = \emptyset$. Consequently $\mathit{inf}(\tilde{s}) = (\mathit{pre}(\tilde{s}), \mathit{post}(\tilde{s}), (\cup_{t \in \tilde{s}} {}^{-}t) \setminus \mathit{post}(\tilde{s}))$ and $\mathit{inf}(m \setminus \mathit{pre}(\tilde{s})) = (m \setminus \mathit{pre}(\tilde{s}), m \setminus \mathit{pre}(\tilde{s}), \emptyset)$ fulfill $(\mathit{inf}(\tilde{s}), \mathit{inf}(m \setminus \mathit{pre}(\tilde{s})) \in \mathit{dom}_{\dot{\|}}$ and thus $\alpha = \tilde{s} \,\|\, (m \setminus \mathit{pre}(\tilde{s}))$ is defined, with $\mathit{pre}(\alpha) = m$ and $\mathit{post}(\alpha) = m'$ ($\mathit{post}(\alpha) = m'$ is a simple calculation). The existence of such α gives the first part of the equivalence in Definition 5.1.

For the second part let $\alpha = \tilde{s} \,\|\, \tilde{m} : m \to m'$ ($\tilde{s} \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}, \tilde{m}, m, m' \in M$) be a defined process term. Since by definition of $\mathit{dom}_{\dot{\|}}$ the sets $\mathit{pre}(\tilde{s})$ and $\tilde{m}$ are disjoint, $\mathit{pre}(\tilde{s}) \cup \tilde{m} = m$ implies $\tilde{m} = m \setminus \mathit{pre}(\tilde{s})$. We show that the step $|\tilde{s}| \in 2^T$ is enabled to occur in m. If not, then either there are transitions $t_1, t_2 \in |\tilde{s}|$ which are in synchronous conflict or there is a transition $t \in |\tilde{s}|$, which is not enabled to occur in m.

Suppose that $t_1, t_2 \in |\tilde{s}|$ are in synchronous conflict. Then (independent from the construction order of $\tilde{s}$) the construction of $\tilde{s}$ involves a sub-term $s' \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ of $\tilde{s}$ and $s_1, s_2 \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ such that $s' = s_1 \oplus s_2$ and t_1 is a sub-term of s_1 and t_2 is a sub-term of s_2. Denoting $\mathit{inf}(s_1) = (a,b,c)$ and $\mathit{inf}(s_2) = (d,e,f)$ the synchronous conflict of t_1 and t_2 yields one of the following non-empty intersections: $(a \cup b) \cap (d \cup e) \neq \emptyset$ or $a \cap f \neq \emptyset$ or $c \cap d \neq \emptyset$. Consequently $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \notin \mathit{dom}_{\dot{\oplus}}$ and thus the process term s' is not defined. A contradiction.

It remains to show that every $t \in |\tilde{s}|$ is enabled to occur in m. First we calculate ${}^\bullet t = \mathit{pre}(t) \subseteq \mathit{pre}(\tilde{s}) \subseteq m$. Second by definition of $\mathit{dom}_{\dot{\|}}$ we can conclude that $t^\bullet \cap \tilde{m} = \emptyset$. Moreover by definition of $\mathit{dom}_{\dot{\oplus}}$ we can conclude that $t^\bullet \cap {}^\bullet u = \emptyset$ for $u \in |\tilde{s}| \setminus \{t\}$ (an analogues argumentation as above that the construction of $\tilde{s}$ involves a sub-term $s' \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ of $\tilde{s}$ and $s_1, s_2 \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ such that $s' = s_1 \oplus s_2$ and t is a sub-term of s_1 and u is a sub-term of s_2). Thus from $\mathit{pre}(\tilde{s}) \cup \tilde{m} = m$ we have $(m \setminus {}^\bullet t) \cap t^\bullet = \emptyset$. Altogether, every $t \in |\tilde{s}|$ is enabled to occur in m.

Consequently we have shown that the step $|\tilde{s}|$ is enabled to occur in m. With $\tilde{m} = m \setminus \mathit{pre}(\tilde{s})$ it is easy to check that its occurrence leads to $\mathit{post}(\alpha) = m'$.

Proof of Lemma 6.2

(Con1): $x, y \in M$: $(\mathit{inf}(x), \mathit{inf}(y)) \in dom_{\dot{\|}} \implies \mathit{inf}(x) \mathbin{\dot{\|}} \mathit{inf}(y) = (x, x, \emptyset) \mathbin{\dot{\|}} (y, y, \emptyset) = (x \cup y, x \cup y, \emptyset) = \mathit{inf}(x \cup y) = \mathit{inf}(x + y)$ (according to the definition of $\dot{\|}$, x and y even have to be disjoint).

(Con2): Obviously $\emptyset$ is the neutral element of $\mathcal{M}$ and $(\emptyset, \emptyset, \emptyset)$ the neutral element of $(I, dom_{\dot{\|}}, \dot{\|})$ and it holds $\mathit{inf}(\emptyset) = (\emptyset, \emptyset, \emptyset)$.

(Con3): $s \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}$: $\mathit{supp}(\{\mathit{inf}(s), \mathit{inf}(\mathit{pre}(s)), \mathit{inf}(\mathit{post}(s))\}) = \mathit{supp}(\{(\mathit{pre}(s), \mathit{post}(s), {}^{-}s \setminus \mathit{post}(s)), (\mathit{pre}(s), \mathit{pre}(s), \emptyset), (\mathit{post}(s), \mathit{post}(s), \emptyset)\}) = (\mathit{pre}(s) \cup \mathit{post}(s), {}^{-}s \setminus (\mathit{pre}(s) \cup \mathit{post}(s))) = \mathit{supp}(\{\mathit{inf}(s)\})$.

(Con4): $s_1, s_2 \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}$ with $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in dom_{\dot{\oplus}}$: $\mathit{supp}(\{\mathit{inf}(s_1 \oplus s_2), \mathit{inf}(s_1), \mathit{inf}(s_2)\}) = (\mathit{pre}(s_1) \cup \mathit{pre}(s_2) \cup \mathit{post}(s_1) \cup \mathit{post}(s_2), ({}^{-}s_1 \cup {}^{-}s_2) \setminus (\mathit{pre}(s_1) \cup \mathit{pre}(s_2) \cup \mathit{post}(s_1) \cup \mathit{post}(s_2))) = \mathit{supp}(\{\mathit{inf}(s_1 \oplus s_2)\})$.

(Con5): $s_1, s_2 \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}, m, m_1, m_2 \in M$: $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in dom_{\dot{\oplus}}$, $(\mathit{inf}(s_1) \mathbin{\dot{\oplus}} \mathit{inf}(s_2), m) \in dom_{\dot{\|}}$, $(\mathit{inf}(s_1), \mathit{inf}(m_1)) \in dom_{\dot{\|}}$, $(\mathit{inf}(s_2), \mathit{inf}(m_2)) \in dom_{\dot{\|}}$, $\mathit{post}(s_1) + m_1 = \mathit{pre}(s_2) + m_2$ and $\mathit{pre}(s_1) + \mathit{pre}(s_2) + m = \mathit{pre}(s_1) + m_1 \implies \mathit{pre}(s_1) \cap \mathit{pre}(s_2) = \mathit{pre}(s_1) \cap m = \mathit{pre}(s_1) \cap m_1 = \emptyset$ (directly from the preliminaries w.r.t. $dom_{\dot{\oplus}}$ and $dom_{\dot{\|}}$) $\implies m_1 = \mathit{pre}(s_2) + m$ (directly from $\mathit{pre}(s_1) + \mathit{pre}(s_2) + m = \mathit{pre}(s_1) + m_1$), consequently we get the first statement $(\mathit{inf}(\mathit{pre}(s_2) + m), \mathit{inf}(s_1)) \in dom_{\dot{\|}}$. Now $\mathit{post}(s_1) + m_1 = \mathit{pre}(s_2) + m_2$ implies $\mathit{pre}(s_2) + \mathit{post}(s_1) + m = \mathit{pre}(s_2) + m_2$. From the preliminaries w.r.t. $dom_{\dot{\oplus}}$ and $dom_{\dot{\|}}$ there onward results $\mathit{pre}(s_2) \cap \mathit{post}(s_1) = \mathit{pre}(s_2) \cap m = \mathit{pre}(s_2) \cap m_2 = \emptyset$ and therefore $m_2 = \mathit{post}(s_1) + m$, consequently we get the second statement $(\mathit{inf}(\mathit{post}(s_1) + m), \mathit{inf}(s_2)) \in dom_{\dot{\|}}$.

(Con6): $s_1, s_2 \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}$: $(\mathit{inf}(s_1), \mathit{inf}(s_2)) \in dom_{\dot{\|}} \implies (\mathit{inf}(s_1), \mathit{inf}(s_2)) \in dom_{\dot{\oplus}}$ (according to the definition) and $\mathit{supp}(\{\mathit{inf}(s_1) \mathbin{\dot{\|}} \mathit{inf}(s_2), \mathit{inf}(s_1) \mathbin{\dot{\oplus}} \mathit{inf}(s_2)\}) = (\mathit{pre}(s_1) \cup \mathit{pre}(s_2) \cup \mathit{post}(s_1) \cup \mathit{post}(s_2), ({}^{-}s_1 \cup {}^{-}s_2) \setminus (\mathit{pre}(s_1) \cup \mathit{pre}(s_2) \cup \mathit{post}(s_1) \cup \mathit{post}(s_2))) = \mathit{supp}(\{\mathit{inf}(s_1) \mathbin{\dot{\|}} \mathit{inf}(s_2)\})$ (it even holds $\mathit{inf}(s_1) \mathbin{\dot{\|}} \mathit{inf}(s_2) = \mathit{inf}(s_1) \mathbin{\dot{\oplus}} \mathit{inf}(s_2)$).

(Det): $s \in \mathit{Step}_{(\mathcal{A}, \mathit{inf})}, x, y \in M$: $(\mathit{inf}(s), \mathit{inf}(x)) \in dom_{\dot{\|}}$, $(\mathit{inf}(s), \mathit{inf}(y)) \in dom_{\dot{\|}}$ and $\mathit{pre}(s) + x = \mathit{pre}(s) + y \implies \mathit{pre}(s) \cap x = \mathit{pre}(s) \cap y = \emptyset$ (from the preliminaries w.r.t. $dom_{\dot{\|}}$) $\implies x = y$ (directly from $\mathit{pre}(s) + x = \mathit{pre}(s) + y$), therefore especially $\mathit{post}(s) + x = \mathit{post}(s) + y$.

Proof of Lemma 6.3

It is enough to consider the cases where α is derived from β through one of the equivalent transformation axioms (1)-(11) (Definition 4.3). Because for axioms preserving associated so-structures the statement is trivial we will only consider the axioms (4) and (7). We will prove the statement by contradiction. Let $AON = (B, V, R, \mathit{Act}, l)$ (with $l|_V = id$) be the process represented by the run $\mathcal{S}$.

First we consider axiom (4). It is enough to consider the case $\alpha = (\alpha_1; \alpha_3) \| (\alpha_2; \alpha_4)$ and $\beta = (\alpha_1 \| \alpha_2); (\alpha_3 \| \alpha_4)$ (since in this case $\mathcal{S}_1 \subseteq \mathcal{S}_2$ because in $\mathcal{S}_2$ orderings ($\prec$ and $\sqsubset$) between events in α_1 and α_4 as well as α_2 and α_3 are added compared to $\mathcal{S}_1$). Without loss of generality, suppose that in the run an ordering between an event in α_1 and an event in α_4 exists ($\prec$ or $\sqsubset$-ordering). That means there are events $t \in \alpha_1$, and $s \in \alpha_4$, and a condition $c \in B$ such that one of the following three possibilities holds (according to Figure 6): (a) $(t, c) \in R$ and $(c, s) \in R$ or (b) $(t, c) \in R$ and $(c, s) \in \mathit{Act}$ or (c) $(c, t) \in \mathit{Act}$ and $(c, s) \in R$.

Consider case (a): If $l(c) = x \in P$ (no complement place), then we have $x \in t^\bullet$, $x \in {}^\bullet s$ and therefore $Inf(\alpha_1) = (w_1, a_1)$, $Inf(\alpha_4) = (w_4, a_4)$ with $x \in w_1 \cap w_4$. This contradicts the fact that $\alpha = (\alpha_1; \alpha_3) \| (\alpha_2; \alpha_4)$ is a defined process term. If $l(c) = x' \in P'$ then $c^{-1}(x') \in {}^\bullet t$ and $c^{-1}(x') \in s^\bullet$ causes the same contradiction.

Consider case (b): We have $l(c) = x' \in P'$ ($l(c)$ has to be a complement place, because c is in Act-relation to an event), then $c^{-1}(x') \in {}^\bullet t$ and $c^{-1}(x') \in {}^- s$ and therefore $Inf(\alpha_1) = (w_1, a_1)$, $Inf(\alpha_4) = (w_4, a_4)$ with $c^{-1}(x') \in w_1$ and $c^{-1}(x') \in w_4 \cup a_4$. This contradicts the fact that $\alpha = (\alpha_1; \alpha_3) \| (\alpha_2; \alpha_4)$ is a defined process term. Case (c) causes a contradiction analogously as in case (b).

Now we check axiom (7). For this axiom we have to discuss the equivalence transformation in both directions. Let first $\alpha = (\alpha_1 \| pre(\alpha_2)); (\alpha_2 \| post(\alpha_1))$ and $\beta = \alpha_1 \oplus \alpha_2$ (α_1 and α_2 have to be synchronous step terms). Suppose that in the run an $\sqsubset$- ordering between an event in α_2 and an event in α_1 exists. That means there are events $s \in \alpha_1$, and $t \in \alpha_2$, and a condition $c \in B$ such that the following relation holds: $(c, t) \in Act$ and $(c, s) \in R$. We have $l(c) = x' \in P'$, then $c^{-1}(x') \in {}^- t$ and $c^{-1}(x') \in s^\bullet \subseteq post(alpha_1)$ and therefore $inf(\alpha_2) = (a_2, b_2, c_2)$, $inf(post(\alpha_1)) = (a_1, b_1, c_1)$ with $c^{-1}(x') \in b_2 \cup c_2$ and $c^{-1}(x') \in a_1 = b_1$. This contradicts the fact that $\alpha_2 \| post(\alpha_1)$ is a defined process term.

Let on the other hand $\beta = (\alpha_1 \| pre(\alpha_2))$; $(\alpha_2 \| post(\alpha_1))$ and $\alpha = \alpha_1 \oplus \alpha_2$. Suppose that in the run an $\prec$-ordering between an event in α_1 and an event in α_2 exists. It means there are events $t \in \alpha_1$, and $s \in \alpha_2$, and a condition $c \in B$ such that one of the following relation holds: (a) $(t, c) \in R$ and $(c, s) \in R$ or (b) $(t, c) \in R$ and $(c, s) \in Act$.

Consider case (a): If $l(c) = x \in P$, then we have $x \in t^\bullet$, $x \in {}^\bullet s$ and therefore $inf(\alpha_1) = (a_1, b_1, c_1)$, $inf(\alpha_2) = (a_2, b_2, c_2)$ with $x \in b_1 \cap a_2$. This contradicts the fact that $\alpha = \alpha_1 \oplus \alpha_2$ is a defined process term. If $l(c) = x' \in P'$ then $c^{-1}(x') \in {}^\bullet t$ and $c^{-1}(x') \in s^\bullet$ causes the same contradiction.

Consider case (b): We have $l(c) = x' \in P'$, then $c^{-1}(x') \in {}^\bullet t$ and $c^{-1}(x') \in {}^- s$ and therefore $inf(\alpha_1) = (a_1, b_1, c_1)$, $inf(\alpha_2) = (a_2, b_2, c_2)$ with $c^{-1}(x') \in a_1$ and $c^{-1}(x') \in c_2 \cup b_2$. This contradicts the fact that $\alpha = \alpha_1 \oplus \alpha_2$ is a defined process term.