Synchronous + Concurrent + Sequential = Earlier than + Not later than

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

Robert Lorenz, Sebastian Mauser
Department of Applied Computer Science
Catholic University of Eichstätt-Ingolstadt
robert.lorenz@ku-eichstaett.de
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 exemplarily for elementary nets with inhibitor arcs.

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. [13, 6, 10, 11]).

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 [18, 1, 3, 5, 16, 14, 15] (see [17] for an overview). Most of these approaches are based on the paper [12], 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 [7]. With the approach from [7] *non-sequential semantics* can be derived on an abstract level for Petri nets with restricted occurrence rule (encoded by partiality of concurrent composition). In addition to other works, and in particular to [5], in [7] it is shown how to obtain causal semantics based on "earlier than" causality between events (formally given as labelled partial orders (LPOs)) from process terms. It is shown in [7] for many concrete net classes that the minimal LPOs obtained from process terms coincide with minimal LPOs given by acknowledged classical processes.

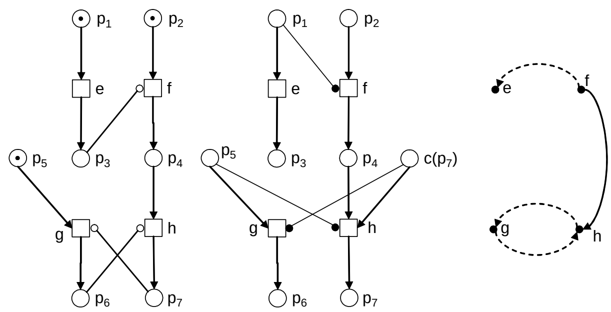

Figure 1. An elementary net with inhibitor arcs (p_3, f), (p_6, h) and (p_7, g), a process of the net and the associated run.

As explained in [6], "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 this phenomenon is depicted: In the a-priori semantics the testing for absence of tokens (through inhibitor arcs) precedes the execution of a transition. Thus f cannot occur later than e, because after the occurrence of e the place p_3 is marked

[1] Note that there are also other semantics for elementary nets with inhibitor arcs such as the a-posteriori semantics which are less problem-prone [6].

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 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 even only occur synchronously but not sequentially in any order. The described causal behavior (between events) of the net is illustrated on the right side of Figure 1 ("run"). The drawn through arcs represent a (common) "earlier than"-relation, i.e. the events can only occur in the expressed order but not synchronously or inversely, and dotted arcs depict the "not later than" relation explained above. The net in the middle of Figure 1 depicts a process corresponding to the run on the right (details on processes and runs are explained in the next section). Altogether there exist net classes including the by practitioners admired inhibitor nets where synchronous and concurrent behaviour has to be distinguished. In [6] causal semantics based on stratified order structures like the run in Figure 1 (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.

In order to describe such situations on the algebraic level, in [8] we extended the algebraic Petri nets from [5] by a synchronous composition operation which allows to distinguish between concurrent and synchronous occurrences of events. Therewith a great variety of additional concrete net classes can be covered compared to [7]. Unfortunately, paper [8] does not provide a general method how to construct so-structure based causal semantics from algebraic semantics. Therefore in [8] a correspondence of the algebraic semantics to non-sequential a-priori 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 causal semantics based on so-structures from process terms and derive exemplarily their correspondence to causal semantics produced from processes for elementary nets with inhibitor arcs equipped with the a-priori semantics.

Thereto we generalize our own algebraic approach from [8] generating process terms via 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 (Section 4). These causal semantics are consistent with the step semantics of algebraic nets in the sense that an so-structure is enabled iff every of its step sequentializations is an enabled step sequence. Given a Petri net of a concrete Petri net class, we define the corresponding algebraic net to have the same step semantics (Section 5). Then for concrete Petri net classes with causal semantics derived from processes minimal enabled so-structures obtained from process terms of a corresponding algebraic net coincide with minimal so-structures given by processes. Exemplarily we will show this result in a systematic way (which can obviously be adapted to further net classes) for elementary nets with inhibitor arcs (Section 6)[2], thus generalizing the main result of [8].

For better readability we only sketched the proofs in the main text of the paper but we included detailed proofs in the Appendix.

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 $R \circ R'$. A *directed graph* is a pair $(V, \to)$, where V is a finite *set of nodes* and $\to \subseteq V \times V$ is a binary relation over V called the *set of arcs*. As usual, given a binary relation $\to$, we write $a \to b$ to denote $(a, b) \in \to$. Two nodes $a, b \in V$ are called *independent* w.r.t. the binary relation $\to$ if $a \not\to b$ and $b \not\to a$. We denote the set of all pairs of nodes independent w.r.t. $\to$ by $\mathrm{co}_\to \subseteq V \times V$. 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*. 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 $<_1 \subseteq <_2$. A *relational structure* (rel-structure) is a triple $\mathcal{S} = (X, \prec, \sqsubset)$, where X is a 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 1 (Stratified order structure [6])**.** *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) $x \prec y \Longrightarrow x \sqsubset y$

[2] This net class has the advantage that it is already extensively analysed in the concept of ad-hoc process definitions so that we are able to check the consistency of the ad-hoc concept to our general algebraic concept. Furthermore this comparison will not be too lengthy because the ad-hoc definitions are not that complicated as for p/t-nets with inhibitor arcs.

(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 drawn through arcs and $\sqsubset$ by dotted arcs. According to (C2) a dotted arc is omitted if there is already a drawn through arc. Moreover, we omit arcs which can be deduced by (C3) and (C4). It is shown in [6] that $(X, \prec)$ is a partial order. Therefore so-structures are a generalization of partial orders which turned out to be adequate to model the causal relations between events of complex systems regarding 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 and 2). The *$\diamondsuit$-closure* of a rel-structure $\mathcal{S} = (X, \prec, \sqsubset)$ is given by $\mathcal{S}^{\diamondsuit} = (X, \prec_{\mathcal{S}^{\diamondsuit}}, \sqsubset_{\mathcal{S}^{\diamondsuit}}) = (X, (\prec \cup \sqsubset)^{*} \circ \prec \circ (\prec \cup \sqsubset)^{*}, (\prec \cup \sqsubset)^{*} \setminus id_X)$. 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 [6] (observe that the notion of the $\diamondsuit$-closure in the context of so-structures corresponds to the concept of the transitive closure in the less general situation of partial orders). Finally, we introduce two subclasses of so-structures which turn out to be associated to (specific subclasses of) process terms of algebraic Petri nets. Let $\mathcal{S} = (X, \prec, \sqsubset)$ be an so-structure, then $\mathcal{S}$ is called *synchronous closed* if $\mathrm{co}_{\prec} = \mathrm{co}_{\sqsubset} \cup (\sqsubset \setminus \prec)$ (e.g. the so-structure in Figure 1 is not synchronous closed) and $\mathcal{S}$ is called *total linear* if $\mathrm{co}_{\prec} = (\sqsubset \setminus \prec) \cup id_X$ (for examples see Figure 2). The set of all total linear extensions (or linearizations) of $\mathcal{S}$ 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 [10][3] shows that every so-structure can be reconstructed from its linearizations (see Figure 2 for an example):

Proposition 2. *Let $\mathcal{S}$ be an so-structure. Then*

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

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 [6] 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 in the following way: A straightforward proof shows that an so-structure $\mathcal{S}$ is synchronous closed if and only if $(\sqsubset \setminus \prec) \cup id_X$ is an equivalence relation. For such so-structures we denote $\sim_{\mathcal{S}} = (\sqsubset \setminus \prec) \cup id_X$, $[x]_{\mathcal{S}} = \{y \in X \mid x \sim_{\mathcal{S}} y\}$ and $X|_{\mathcal{S}} = \{[x]_{\mathcal{S}} \mid x \in X\}$.

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

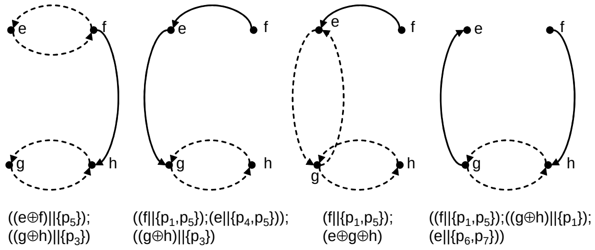

Figure 2. Top: All linearizations of the so-structure from Figure 1. Bottom: Process terms of the net from Figure 1, to which the respective so-structures above are associated.

The elements of $X|_{\mathcal{S}}$ are called *synchronous classes of $\mathcal{S}$*. For $[x]_{\mathcal{S}}, [y]_{\mathcal{S}} \in X|_{\mathcal{S}}$ define $[x]_{\mathcal{S}} <_{\mathcal{S}} [y]_{\mathcal{S}} \iff x \prec y$ (by (C4) this is well-defined), then $(X|_{\mathcal{S}}, <_{\mathcal{S}})$ defines a partial order. The partial order $po_{\mathcal{S}} = (X|_{\mathcal{S}}, <_{\mathcal{S}})$ is called *associated to $\mathcal{S}$*. The partial orders associated to the total linear so-structures in Figure 2 are the total orders expressed by the following sequences (from left to right): $\{e, f\} \rightarrow \{g, h\}$, $\{e\} \rightarrow \{f\} \rightarrow \{g, h\}$, $\{f\} \rightarrow \{e, g, h\}$ and $\{f\} \rightarrow \{g, h\} \rightarrow \{e\}$ (this also illustrates the second statement in Remark 3). We have the following results for associated partial orders:

Remark 3. *Let $\mathcal{S}, \mathcal{S}'$ be synchronous closed so-structures satisfying $\mathcal{S} \subseteq \mathcal{S}'$. Then obviously $k' \in X|_{\mathcal{S}'}$ has the form $k' = k_1 \cup \ldots \cup k_n$ with $k_1, \ldots, k_n \in X|_{\mathcal{S}}$ because $\sim_{\mathcal{S}} \subseteq \sim_{\mathcal{S}'}$.*

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

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* $l : X \rightarrow M$.

Next we present the example net class considered 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 and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation. For $x \in P \cup T$ we abbreviate $^{\bullet}x = \{y \in P \cup T \mid (y, x) \in F\}$ (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 $m \subseteq P$ is called a *marking*. A transition $t \in T$ is *enabled to occur* in a marking m of N iff $^{\bullet}t \subseteq m \wedge (m \setminus {^{\bullet}t}) \cap t^{\bullet} = \emptyset$. 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* iff $({^{\bullet}t_1} \cup t_1^{\bullet}) \cap ({^{\bullet}t_2} \cup t_2^{\bullet}) \neq \emptyset$.

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* satisfying $(F \cup F^{-1}) \cap C_{-} = \emptyset$. In the example net of Figure

1 the negative context relation is depicted through so called inhibitor arcs with circles as arrowheads. 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 iff it is enabled to occur in the underlying elementary net (P,T,F) and $^{-}t \cap m = \emptyset$. 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 N iff 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 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' (see the Introduction for an example on the occurrence rule).

Now we introduce the "classical" process semantics for ENI as presented in [6]. 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. It is shown in [13] 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 [4] will be denoted by P' and the complementation-bijection from P to P' will be denoted by c. The processes of ENI will be defined endowing processes of "ordinary" elementary nets (defined as usual by occurrence nets using complementation, see e.g. [8]) with read arcs (also called activator arcs in [6, 10, 11]).

Definition 4 (Activator process)**.** *A* labelled activator occurrence net *(ao-net) is a five-tuple* $AON = (B, E, R, Act, l)$ *satisfying:* (B, E, R) *is an occurrence net,* (B, E, R, Act) *is an elementary net with positive context, and the relational structure*

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

is $\diamondsuit$*-acyclic. An ao-net* AON *is an* activator process *of* $ENI = (P, T, F, C_{-})$ *w.r.t.* m *iff:*

- $ON = (B, E, R, l)$ *is a process of the elementary net* $N = (P, T, F)$ *w.r.t.* m*, and*
- $\forall b \in B, \forall e \in E : (b, e) \in Act \iff (c^{-1}(l(b)), l(e)) \in C_{-}$.

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

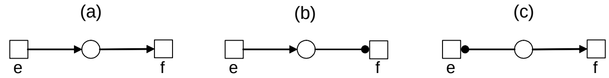

Figure 3. The nets in (a) and (b) generate the order $e \prec f$, the net in (c) the order $e \sqsubset f$.

An example of an activator process and an associated run is depicted in Figure 1. The construction rule of $\mathcal{S}(AON)$ is illustrated in Figure 3. For a more detailed definition of activator processes and a discussion of related results see the series of papers [6, 10, 11].

The central idea to model restricted occurrence rules as in the case of inhibitor nets on the algebraic level is the utilisation of partial algebras 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 $\mathcal{I}$. $\mathcal{I}$ 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 $(a \dot{+} b) \dot{+} c$ is defined then also $a \dot{+} (b \dot{+} c)$ is defined with $(a \dot{+} b) \dot{+} c = a \dot{+} (b \dot{+} c)$ (*closed associativity*) and there is a (unique) *neutral element* $i \in I$ such that $a \dot{+} i$ is defined for all $a \in I$ with $a \dot{+} i = a$ (*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 $\sim$. For this, denote $[x]_{\sim} = \{y \in X \mid x \sim y\}$ and $X/_{\sim} = \{[x]_{\sim} \mid x \in X\}$, $dom_{op/_{\sim}} = \{([a_1]_{\sim}, \ldots, [a_n]_{\sim}) \mid (a_1, \ldots, a_n) \in dom_{op}\}$ 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 alge-

[4] 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.

bra $\mathcal{X}/_\sim$ 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* [2]. The most important result of [2] for this paper is that there always exists a unique greatest closed congruence on a given partial algebra.

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

A *general algebraic Petri net* is a quadruple $\mathcal{A} = (M, T, pre\colon T \to M, post\colon T \to M)$ (similar to [12]), which is a graph with vertices representing *markings* and edges labelled by *transitions*. Formally, the set of markings is given by a (total) commutative monoid $\mathcal{M} = (M, +)$ with neutral element $\underline{0}$. T denotes the set of transitions. The two mappings $pre\colon T \to M, post\colon T \to M$ assign *pre-sets* and *post-sets* to each transition. In order to obtain process term semantics, firstly transitions can be synchronously composed to *synchronous step terms*, and secondly markings and synchronous step terms can be sequentially and concurrently composed to *process terms*. As usual, 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 partial operations $\dot{\|} : dom_{\|} \to I$ and $\dot{\oplus} : dom_{\oplus} \to I$, $dom_{\|}, dom_{\oplus} \subseteq I \times I$, for the concurrent and synchronous composition of information elements, resulting in a partial algebra $\mathcal{I} = (I, dom_{\|}, \dot{\|}, dom_{\oplus}, \dot{\oplus})$. The groupoids $(I, dom_{\|}, \dot{\|})$ and $(I, dom_{\oplus}, \dot{\oplus})$ are assumed to be partial closed commutative monoids with neutral elements i_0 and j_0. Such $\mathcal{I}$ is called *sc-partial algebra*.

Definition 5 (Algebraic $(\mathcal{M}, \mathcal{I})$-net)**.** *Let* $\mathcal{I} = (I, dom_{\|}, \dot{\|}, dom_{\oplus}, \dot{\oplus})$ *be an sc-partial algebra,* $\mathcal{A} = (M, T, pre\colon T \to M, post\colon T \to M)$ *be a general algebraic Petri net, and* $inf : M \cup T \to I$ *be a mapping. Then* $(\mathcal{A}, inf)$ *is called* algebraic $(\mathcal{M}, \mathcal{I})$-net.

For the example (Figure 1) of elementary nets with inhibitor arcs the crucial mappings pre, $post$ and inf for a corresponding algebraic $(\mathcal{M}, \mathcal{I})$-net (see Section 5 for the technical definition of corresponding) will be defined as follows: $pre(t) = {}^\bullet t$, $post(t) = t^\bullet$, $inf(t) = ({}^\bullet t, t^\bullet, {}^- t)$ for $t \in T$ and $inf(m) = (m, m, \emptyset)$ for $m \subseteq P$ (see Section 6 for details), i.e. $\mathcal{M} = (2^P, \cup)$, $T = T$ and $I = 2^P \times 2^P \times 2^P$. The first two components of $i \in I$ represent the write part - pre and $post$ - and the last component stores the read information - the negative context which is not in the write part. Consequently for the example net from Figure 1 we have $inf(e) = (\{p_1\}, \{p_3\}, \emptyset)$, $inf(f) = (\{p_2\}, \{p_4\}, \{p_3\})$, $inf(g) = (\{p_5\}, \{p_6\}, \{p_7\})$, $inf(h) = (\{p_4\}, \{p_7\}, \{p_6\})$. Note that in this example it is now important which information triples can be composed synchronously respectively concurrently and which information triples result from such a composition. Thereto completely coincident with the occurrence rule of elementary nets with inhibitor arcs equipped with the a-priori semantics, two information elements $i_1, i_2 \in I$ can be composed

- concurrently iff the write part of i_1 is disjoint from the write and the read part of i_2 and vice versa,
- synchronously iff the write parts of i_1 and i_2 are disjoint and the pre-component (first component) of i_1 is disjoint from the read part of i_2 and vice versa.

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 their information elements. Accordingly in our 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. Thus, in general the synchronous step terms of an algebraic $(\mathcal{M}, \mathcal{I})$-net are defined inductively as follows.

Definition 6 (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* $(inf(s), inf(s')) \in dom_{\oplus}$, *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* $Step_{(\mathcal{A}, inf)}$.

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 will have 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. 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 $post$ of the first process term has to coincide with pre of the second process term while for concurrent composability the information sets of the two process terms have to be independent [5]. Consequently some possible process terms in the example are $f;h$, $e;\{p_3\}$, $e \| g$, $(e \oplus f) \| g$, $(g \oplus h) \| \{p_3\}$ or $((e \oplus f) \| \{p_5\}); ((g \oplus h) \| \{p_3\})$ (with the last example term the basic principle of constructing process terms is illustrated in Figure 4). For the technical definitions the sc-partial algebra $\mathcal{I} = (I, dom_{\dot{\|}}, \dot{\|}, dom_{\dot{\oplus}}, \dot{\oplus})$ is lifted to the partial algebra $\mathcal{X} = (2^I, dom_{\{\dot{\|}\}}, \{\dot{\|}\}, 2^I \times 2^I, \cup)$ defined by $dom_{\{\dot{\|}\}} = \{(X,Y) \in 2^I \times 2^I \mid X \times Y \subseteq dom_{\dot{\|}}\}$ and $X\{\dot{\|}\}Y = \{x \,\dot{\|}\, y \mid x \in X \wedge y \in Y\}$. It is easy to verify that $\mathcal{X}$ is also a partial closed commutative monoid. 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 $\mathcal{X}$ (for a concrete construction of $\cong$ see Section 6). 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 7 (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 $Inf(\alpha;\beta) = Inf(\alpha) \cup/_\cong Inf(\beta)$.

[5]Two information sets X and Y are called *independent* if each information element in X is independent from (i.e. concurrently composable with) each information element in Y.

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

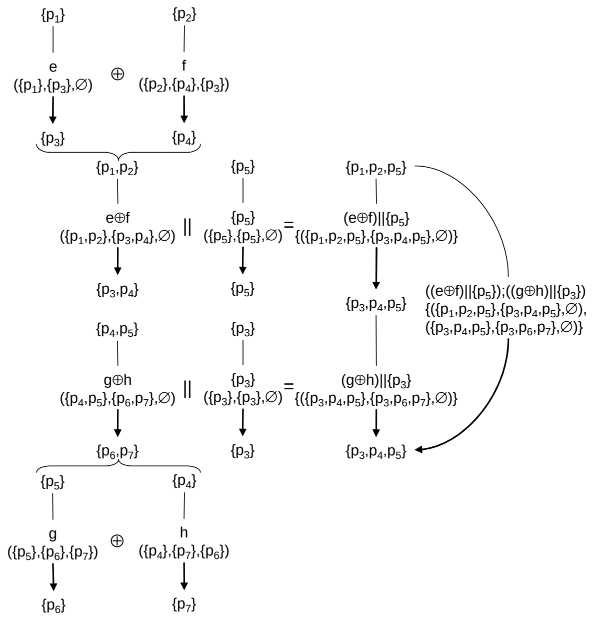

Figure 4. Deriving exemplary a process term of the net from Figure 1. In the middle of an arc there are drawn the respective subprocess terms with associated information (information elements for elementary process terms and information sets for non-elementary process terms) in the line below. At the beginning of an arrow we illustrated pre **and at the arrowhead** $post$ **is depicted.**

For process terms 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 4 are $e \oplus f$ and $g \oplus h$. For a process term α and $t \in T$ we denote $t \in \alpha \iff (\exists s \in B_\alpha : t \in s)$.

Compared to [8] and [7], the definition of algebraic $(\mathcal{M}, \mathcal{I})$-nets is as general as possible. In order to derive conclusions about process term semantics on the algebraic level similar as in [7] 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 $(\mathcal{I}, dom_{\dot{\|}}, \dot{\|})$, $(\mathcal{I}, dom_{\dot{\oplus}}, \dot{\oplus})$ and $\mathcal{M} = (M, +)$. All properties have a simple intuitive interpretation and for all common net classes (with so-structure based semantics) it is easy to show that they are fulfilled. In contrast to [8] 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) $(inf(x), inf(y)) \in dom_{\dot{\|}} \implies inf(x+y) = inf(x) \,\dot{\|}\, inf(y)$ (consistency of markings)

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

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

(Con4) $(inf(s_1), inf(s_2)) \in dom_{\dot{\oplus}} \implies \{inf(s_1 \oplus s_2)\} \cong \{inf(s_1 \oplus s_2), inf(s_1), inf(s_2)\}$ (consistency of steps)

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

(Con6) $(inf(s_1), inf(s_2)) \in dom_{\dot{\|}} \implies (inf(s_1), inf(s_2)) \in dom_{\dot{\oplus}}$ and $\{inf(s_1) \,\dot{\|}\, inf(s_2)\} \cong \{inf(s_1) \,\dot{\|}\, inf(s_2), \; inf(s_1) \,\dot{\oplus}\, inf(s_2)\}$ (synchronous consistency)

(Det) $(inf(s), inf(x)), (inf(s), inf(y)) \in dom_{\dot{\|}}$, $pre(s) + x = pre(s) + y \implies post(s) + x = post(s) + y$ (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 $pre(s)$ and $post(s)$ 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 subsection by enumerating some technical notions and immediate observations concerning the greatest closed congruence $\cong$ on $\mathcal{X}$. For $A \in 2^I$ 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 sense that 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 $a \subseteq /_{\cong} b$ if there exist $A, B \in 2^I$ with $A \subseteq B$, $a = [A]$ and $b = [B]$ $(a, b \in 2^I/_{\cong})$. Simple technical computations yield for $a, b, a', b' \in 2^I/_{\cong}$:
(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 $\implies a \{\dot{\|}\}/_{\cong} b$ defined.
(iii) $a \subseteq /_{\cong} a'$, $b \subseteq /_{\cong} b' \implies (a \cup /_{\cong} b) \subseteq /_{\cong} (a' \cup /_{\cong} b')$ 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) $a \subseteq /_{\cong} c$, $b \subseteq /_{\cong} c \implies (a \cup /_{\cong} b) \subseteq /_{\cong} c$.

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

We define explicit causal semantics of algebraic $(\mathcal{M}, \mathcal{I})$-nets by associating so-structures to process terms.

Definition 8 (So-structures of process terms). *We define inductively* labelled so-structures $go_\alpha = (V, \prec_\alpha, \sqsubset_\alpha, l_\alpha)$ of (or associated to) a process terms α*:* $go_m = (\emptyset, \emptyset, \emptyset, \emptyset)$ *for* $m \in M$*,* $go_t = (\{v\}, \emptyset, \emptyset, l)$ *with* $l(v) = t$ *for* $t \in T$*, and* $go_{s_1 \oplus s_2} = (V_1 \cup V_2, \emptyset, \sqsubset_1 \cup \sqsubset_2 \cup (V_1 \times V_2) \cup (V_2 \times V_1), l_1 \cup l_2)$*, for synchronous step terms* $s_1, s_2 \in Step_{(\mathcal{A}, inf)}$ *with associated so-structures* $go_1 = (V_1, \emptyset, \sqsubset_1, l_1)$ *and* $go_2 = (V_2, \emptyset, \sqsubset_2, l_2)$*, 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* $go_1 = (V_1, \prec_1, \sqsubset_1, l_1)$ *and* $go_2 = (V_2, \prec_2, \sqsubset_2, l_2)$*, define*

- $go_{\alpha_1 \,\|\, \alpha_2} = (V_1 \cup V_2, \prec_1 \cup \prec_2, \sqsubset_1 \cup \sqsubset_2, l_1 \cup l_2)$,
- $go_{\alpha_1 ; \alpha_2} = (V_1 \cup V_2, \prec_1 \cup \prec_2 \cup (V_1 \times V_2), \sqsubset_1 \cup \sqsubset_2 \cup (V_1 \times V_2), l_1 \cup l_2)$,

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

Since all labelled so-structures associated to a process term α are isomorphic (and arbitrary labelled so-structures isomorphic to go_α 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 go_α of a process term α is synchronous closed.

In Figure 2 the principle of so-structures associated to process terms is demonstrated. Note that there cannot exist a process term to which the run from Figure 1 is associated because this so-structure is not synchronous closed. That is why we considered its linearizations (which are always synchronous closed) here. 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.

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 a process term in general cannot describe each run of a Petri net (e.g. as explained the run from Figure 1 is not associated to a process term; other examples which are also valid for the partial order case include so called N-forms [7]). 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 [7] 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*, and the set of all synchronous step sequence terms with initial marking m by $Stepseq_{(\mathcal{A}, inf, m)}$. It is easy to observe that so-structures associated to synchronous step sequence terms are total linear.

Definition 9 (Enabled labelled so-structure)**.** *A labelled so-structure go is* enabled to occur *in a marking m w.r.t. an algebraic $(\mathcal{M}, \mathcal{I})$-net $(\mathcal{A}, inf)$, if every $go' \in strat_{sos}(go)$ is associated to some $\beta \in Stepseq_{(\mathcal{A}, inf, m)}$. 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 2 gives an example how to check if an so-structure is enabled. It shows that the run from Figure 1 is enabled w.r.t. the marked net in the same figure. We will show in Theorem 13, 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. A labelled so-structure go enabled in m is said to be *minimal*, if there exists no labelled so-structure $go' \neq go$ enabled in m, where go is an extension of go'. We denote by $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$ the set of all such minimal enabled labelled so-structures. For example with this definition one can check (intuitively and technically) that the run from Figure 1 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 17 that two process terms are equivalent if their associated so-structures can be identified in the above described sense. In this context the process terms in Figure 2 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 [7]). In the example all process terms in Figure 2 represent the (commutative) process in Figure 1.

Definition 10 (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 $\oplus$, $\|$ and ;*[6]*, 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) $\alpha \,\|\, \beta \sim \beta \,\|\, \alpha$
(2) $(\alpha \,\|\, \beta) \,\|\, \gamma \sim \alpha \,\|(\beta \,\|\, \gamma)$
(3) $(\alpha; \beta); \gamma \sim \alpha; (\beta; \gamma)$
(4) $\alpha = ((\alpha_1 \,\|\, \alpha_2); (\alpha_3 \,\|\, \alpha_4)) \sim \beta = ((\alpha_1; \alpha_3) \,\|(\alpha_2; \alpha_4))$
(5) $\alpha \oplus \beta \sim \beta \oplus \alpha$
(6) $(\alpha \oplus \beta) \oplus \gamma \sim \alpha \oplus (\beta \oplus \gamma)$
(7) $(\alpha \oplus \beta) \sim (\alpha \,\|\, pre(\beta)); (post(\alpha) \,\|\, \beta)$
(8) $(\alpha; post(\alpha)) \sim \alpha \sim (pre(\alpha); \alpha)$
(9) $id_{(m+n)} \sim id_m \,\|\, id_n$
(10) $pre(\alpha) + m = pre(\alpha) + n$, $post(\alpha) + m = post(\alpha) + n$ $\Longrightarrow (\alpha \,\|\, id_m) \sim (\alpha \,\|\, id_n)$
(11) $(\alpha \,\|\, id_{\underline{0}}) \sim \alpha$
if the terms on both sides of $\sim$ are defined process terms.

E.g. for the first two process terms in Figure 2 we have the following transformation:

$((e \oplus f) \,\|\{p_5\}); ((g \oplus h) \,\|\{p_3\}) \overset{(7)}{\sim}$
$(((f \,\|\{p_1\}); (e \,\|\{p_4\})) \,\|\{p_5\}); ((g \oplus h) \,\|\{p_3\}) \overset{(8)}{\sim}$
$(((f \,\|\{p_1\}); (e \,\|\{p_4\})) \,\|(\{p_5\}; \{p_5\})); ((g \oplus h) \|\{p_3\}) \overset{(4)}{\sim}$
$((f \,\|\{p_1\} \,\|\{p_5\}); (e \,\|\{p_4\} \,\|\{p_5\})); ((g \oplus h) \,\|\{p_3\}) \overset{(9)}{\sim}$
$((f \,\|\{p_1, p_5\}); (e \,\|\{p_4, p_5\})); ((g \oplus h) \,\|\{p_3\})$ (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

[6] According to [2] this least congruence exists uniquely.

through the axioms (4) and (7). Regarding (4) we get that go_α is an extension of go_β with additional $\prec$-ordering between events of α_1 and α_4 as well as between events of α_2 and α_3. Regarding (7) the associated so-structures are not comparable in a similar way. For further observations concerning the relationship between the above axioms, the properties (Con1)-(Con6) and (Det), and the information associated to process terms see the beginning of the Appendix.

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 go_α 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 go_α since $\{|s| \mid s \in B_\alpha\} = V|_{go_\alpha}$ (see Section 2 for the definition of $V|_{go_\alpha}$). The synchronous class corresponding to $s \in B_\alpha$ is denoted by $k_s \in V|_{go_\alpha}$. 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: For a set V and a surjective labelling function $l : V \to T$ we denote by $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ the algebraic $(\mathcal{M}, \mathcal{I})$-net given by $\mathcal{A}_{(V,l)} = (M, V, pre_{(V,l)}, post_{(V,l)})$ and $\mathit{inf}_{(V,l)} : M \cup V \to I$, where $pre_{(V,l)}(v) = pre(l(v))$, $post_{(V,l)}(v) = post(l(v))$, $\mathit{inf}_{(V,l)}|_M = \mathit{inf}|_M$ 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 13 (and the precursory Lemma 11). 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 $s : C \to X$ we define inductively the *substituted term* α_s of α w.r.t. s: If $\alpha = m \in M$, then $\alpha_s = m$. If $\alpha \in Step_{(\mathcal{A},\mathit{inf})}$, then $\alpha_s = \alpha$ if $\alpha \notin C$ and $\alpha_s = s(\alpha)$ if $\alpha \in C$. Finally, for process terms α, β, γ: If $\alpha = \beta; \gamma$, then $\alpha_s = \beta_s; \gamma_s$ and if $\alpha = \beta \parallel \gamma$, then $\alpha_s = \beta_s \parallel \gamma_s$. Later we will be interested in the case that basic step terms are substituted by their post-sets.

Lemma 11. *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 $go_\alpha = (V, \prec, \sqsubset, id)$ be a labelled so-structure of α with associated partial order $po_{go_\alpha} = (K_{go_\alpha}, <_{go_\alpha})$ and $C \subseteq B_\alpha$ be a set of minimal basic step terms of α (i.e. k_s is minimal w.r.t. $<_{go_\alpha}$ for every $s \in C$)[7]. 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 $(\mathcal{A}, \mathit{inf})$[8] satisfying: (I) $\alpha_C \sim \alpha$ and (II) $\mathit{Inf}(\alpha_C) \subseteq/_{\cong} \mathit{Inf}(\alpha)$.

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

Corollary 12. *Assume the same preconditions as in Lemma 11 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) $\alpha_{Sy} \sim \alpha$ and (II) $\mathit{Inf}(\alpha_{Sy}) \subseteq/_{\cong} \mathit{Inf}(\alpha)$.

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

Theorem 13. *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 $go_\alpha \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ with $m = pre(\alpha)$. In particular, every $go' \in strat_{sos}(go_\alpha)$ is associated to some $\beta \in Stepseq_{(\mathcal{A},\mathit{inf},m)}$ satisfying $\alpha \sim \beta$.*

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 $go' \in strat_{sos}(go_\alpha)$ and satisfies $\alpha \sim \beta$. Such β can be constructed by an iterative application of Corollary 12. See the Appendix for a detailed proof. □

An enabled so-structure go is uniquely determined by the set of process terms whose associated so-structures extend go. As we have already seen in the recurring example the run (an enabled so-structure) from Figure 1 can be reconstructed with the linearizations from Figure 2 which are all associated to certain process terms.

Definition 14. *Let $go = (V, \prec, \sqsubset, l) \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. Then the set Υ_{go}^{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 go.*

Remark 15. *Let Υ_{go}^{can} be the canonical set of the so-structure $go = (V, \prec, \sqsubset, l) \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$. For $\alpha \in \Upsilon_{go}^{can}$ denote $go_\alpha = (V, \prec_\alpha, \sqsubset_\alpha, id)$ the so-structure associated to α. Then $go = (V, \bigcap_{\alpha \in \Upsilon_{go}^{can}} \prec_\alpha, \bigcap_{\alpha \in \Upsilon_{go}^{can}} \sqsubset_\alpha, l)$ by Proposition 2.*

The set Υ_{go}^{can} is maximal 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 the equation there holds $\Upsilon \subseteq \Upsilon_{go}^{can}$.

The next 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

[7] Corresponding synchronous classes k_s of basic step terms s are defined because the labelling function of go_α is id.

[8] Note: α_{su} is the substituted term of α w.r.t. su.

$\sim$-equivalent. In the subsequent theorem we can generalize this result omitting the presumption of total linear so-structures.

Lemma 16. *Let* $(\mathcal{A}, inf)$ *be an algebraic* $(\mathcal{M}, \mathcal{I})$*-net fulfilling (Det) and (Con1)-(Con5). Let* go' *and* go'' *be total linear labelled so-structures of process terms* α *and* β *with initial marking* m*. If* go' *and* go'' *are extensions of* $go \in \mathbf{Enabled}(\mathcal{A}, inf, m)$*, then there holds* $\alpha \sim \beta$*.*

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 *go*). Namely, it is shown that the minimal events of *go* 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 *go*. See the Appendix for a detailed proof. □

Theorem 17. *Let* $(\mathcal{A}, inf)$ *be an algebraic* $(\mathcal{M}, \mathcal{I})$*-net with the same preconditions as in Lemma 16. Let* go' *and* go'' *be labelled so-structures of process terms* α *and* β *with initial marking* m*. If* go' *and* go'' *are extensions of* $go \in \mathbf{Enabled}(\mathcal{A}, inf, m)$*, then there holds* $\alpha \sim \beta$*.*

Proof. There exist total linear so-structures go'_E respectively go''_E which are extensions of go' respectively go''. According to Theorem 13 there exist synchronous step sequence terms α' and β' with associated so-structures go'_E resp. go''_E with $\alpha' \sim \alpha$ and $\beta' \sim \beta$. Clearly, go'_E and go''_E are also extensions of *go*. Thus, according to Lemma 16, 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 15).

Corollary 18. *Let* $(\mathcal{A}, inf)$ *be an algebraic* $(\mathcal{M}, \mathcal{I})$*-net with the same preconditions as in Theorem 17 and let* $go \in \mathbf{Enabled}(\mathcal{A}, inf, m)$*. Then* $\Upsilon_{go}^{can} \subseteq [\alpha]_\sim$ *for some process term* α *of* $(\mathcal{A}_{(V,l)}, inf_{(V,l)})$*. If* $\Upsilon_{go}^{can} = [\alpha]_\sim$*, then* $go \in \mathbf{MinEnabled}(\mathcal{A}, inf, m)$*.*

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

Despite the differences between different classes of Petri nets, there are some common features of almost all net classes, such as the notions of *marking* (state), *transition*, and *occurrence rule* (see [4]). 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 are interested in their concurrent behavior can be obtained from the sequential and synchronous behavior 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 $m \xrightarrow{s} m'$ for $(m, s, m') \in E$ means that s can occur in m with follower marking m'. The notation $m_0 \xrightarrow{s_1 \ldots s_n} m_n$ means that there exist $m_1, \ldots, m_{n-1} \in M$, such that $m_0 \xrightarrow{s_1} m_1, \ldots, m_{n-1} \xrightarrow{s_n} m_n$.

Definition 19 (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* $((M, T, pre\colon T \to M, post\colon T \to M), inf) = (\mathcal{A}, inf)$ *is called a* corresponding net *to* N *if the occurrence rule for synchronous steps is preserved, i.e. if for every pair of markings* $m, m' \in M$ *and every synchronous step* $s \in \mathbb{N}^T$ *there holds:* $m \xrightarrow{s} m'$ *if and only if there exists* $\tilde{s} \in Step_{(\mathcal{A}, inf)}$ *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* $pre(\alpha) = m$ *and* $post(\alpha) = m'$*.*

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

The construction of corresponding algebraic $(\mathcal{M}, \mathcal{I})$-nets provides a general framework to derive causal semantics for a wide range of concrete net classes. 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 exemplary 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 $(\mathcal{A}, inf)$ through defining $\mathcal{M}$, $\mathcal{I}$, *pre*, *post*, *inf* appropriately and deduce $\mathcal{X}$, $\cong$ and a partial algebra of information isomorphic to $\mathcal{X}/_{\cong}$. (3) Show that $(\mathcal{A}, inf)$ 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 $(\mathcal{A}, inf)$ through process terms and thus causal semantics of N through $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$.

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 13): *go* is associated to $m \stackrel{s_1 \dots s_n}{\longrightarrow} m' \iff go \in strat_{sos}(go')$ for some $go' \in \mathbf{MinEnabled}(\mathcal{A}, inf, m)$. 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 $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$. In [7] this was already demonstrated for several Petri net classes with partial order based semantics. 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 $\mathbf{MinEnabled}(\mathcal{A}, inf, m)$. For exotic net classes where their concurrent behaviour cannot be derived from their synchronous and sequential behaviour, the presented framework can be adapted by explicitly regarding the concurrent behaviour in the definition of corresponding algebraic $(\mathcal{M}, \mathcal{I})$-nets[9].

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 recurring example net in Figure 1. 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 [8]. But in [8] 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 using the general framework from Section 5 that can also be adapted to other net classes.

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 [7] by

- $\mathcal{M} = (2^P, \cup)$, $I = 2^P \times 2^P \times 2^P$, $pre(t) = {}^\bullet t$, $post(t) = t^\bullet$, $inf(t) = ({}^\bullet t, t^\bullet, {}^- t)$ $(t \in T)$ and $inf(m) = (m, m, \emptyset)$ $(m \in M)$.
- $dom_{\dot\oplus} = \{((a,b,c),(d,e,f)) \in I \times I \mid (a \cup b) \cap (d \cup e) = a \cap f = d \cap c = \emptyset\}$ 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 $(a,b,c) \dot\| (d,e,f) = (a \cup d, b \cup e, c \cup f)$

and define

- $supp : 2^I \to 2^P \times 2^P$, $supp(A) = (s_1(A), s_2(A) \setminus s_1(A))$ 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 [7] it was shown that $\cong$ is the greatest closed congruence on $\mathcal{X} = (2^I, \{\dot\|\}, dom_{\{\dot\|\}}, 2^I \times 2^I, \cup)$. It is a straightforward computation that the algebraic $(\mathcal{M}, \mathcal{I})$-net defined in this section fulfills all formulated properties of the mapping inf. It is shown in [7]:

Theorem 20. *The algebraic $(\mathcal{M}, \mathcal{I})$-net $((2^P, T, pre, post), inf)$ with $\mathcal{M}, \mathcal{I}, pre, post, inf$ as developed in this section is corresponding to ENI (according to Definition 19).*

To prove the consistency of the algebraic approach to the process based concept we can use an important result about activator processes. Corollary 2 in [10] (considering the more general case of p/t-nets with inhibitor arcs) reads in our terminology:

Theorem 21. $\{go_\alpha \mid \alpha \in Stepseq_{(\mathcal{A}, inf, m)}\} = \bigcup_{r \in \mathbf{Run}(ENI, m)} strat_{sos}(r)$.

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

Lemma 22. *Let $go_1 = (V, \prec_1, \sqsubset_1, id)$ and $go_2 = (V, \prec_2, \sqsubset_2, id)$ be labelled so-structures of $\sim$-equivalent process terms $\alpha : m \to m'$ and $\beta : m \to m'$ of $(\mathcal{A}, inf)$ and let $go = (V, \prec, \sqsubset, id) \in \mathbf{Run}(ENI, m)$ satisfying that $go \subseteq go_2$, then $go \subseteq go_1$.*

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 10). 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 go_1 does not extend go contradicts that α is a defined process term. This can in each case easily be computed by reducing the proof to one of the three situations shown in Figure 3. See the Appendix for a detailed argumentation. □

[9] In this case also the definition of $\mathbf{Enabled}(\mathcal{A}, inf, m)$ has to be adapted accordingly.

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 18). For the reverse statement observe that for $go \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ every so-structure $go' \in strat_{sos}(go)$ is associated to $\alpha \in Stepseq_{(\mathcal{A}, \mathit{inf}, m)}$. All these process terms α are $\sim$-equivalent (Theorem 17) and all elements of $strat_{sos}(go)$ are extensions of one run $r \in \mathbf{Run}(ENI, m)$ (Theorem 21, Lemma 22). Using the representation of go from Proposition 2, we get that go itself is an extension of r. This gives altogether

Theorem 23. *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 13, Remark 15). Consequently there holds the following one-to-one relationship, which is an enhancement of the main result of [8] (proven in another manner).

Theorem 24. *Given ENI and $(\mathcal{A}, \mathit{inf})$ as defined above, the mapping $\psi : \mathbf{Run}(ENI, m) \to \{[\alpha]_\sim \mid pre(\alpha) = m\}$ defined by $\psi(r) = [\alpha]_\sim$ for some α such that go_α is an extension of r is well-defined and bijective.*

Finally, this especially implies that every $go \in \mathbf{Enabled}(\mathcal{A}, \mathit{inf}, m)$ 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. While in other approaches to unifying Petri nets (see e.g. [16, 14, 15, 9]) the occurrence rule is never a parameter and therefore the definitions in [14] and [9] both capture elementary nets but let open more complicated restrictions of enabling conditions in the occurrence rule, such as inhibitor arcs or capacities, in our case we were even able to extend the basic approach from [7] to so-structure based semantics. Thus 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. 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 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,[10] nets with priorities or p/t-nets with weak capacities regarding explicit synchronous semantics.[11] On the other hand it would be interesting to derive behavioral 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] R. Bruni and V. Sassone Algebraic Models for Contextual Nets. *Proc. of ICALP 2000*, Springer, LNCS 1853, pp. 175–186, 2000.

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

[3] A. Corradini P. Baldan and U. Montanari. Contextual petri nets, asymmetric event structures, and processes. *Information and Computation*, 171(1):1–49, 2001.

[4] J. Desel and G. Juhás. What is a Petri Net? LNCS 2128, pages 1-25, 2001.

[5] J. Desel, G. Juhás and R. Lorenz. *Petri Nets over Partial Algebra.* LNCS 2128, pages 126-172, 2001.

[6] R. Janicki and M. Koutny. Semantics of Inhibitor Nets. *Information and Computations* 123, pages 1–16, 1995.

[7] G. Juhás. *Are these events independent? It depends!.* Habilitation thesis, Catholic University Eichstätt-Ingolstadt, 2005.

[8] G. Juhás, R. Lorenz, T. Singliar. *On Synchronicity and Concurrency in Petri Nets.* LNCS 2679, pages 357-376, 2003.

[9] E. Kindler, M. Weber. The Dimensions of Petri Nets: The Petri Net Cube. *EATCS Bulletin* 66, pages 155-166, 1998.

[10] H.C.M. Kleijn and M. Koutny. Process semantics of P/T-Nets with inhibitor arcs. LNCS 1825, pages 261-281, 2000.

[11] H.C.M. Kleijn M. and Koutny. Process semantics of general inhibitor nets. *Inf. and Comp.* 190(1), pages 18-69, 2004.

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

[13] U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6):545–596, 1995.

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

[15] J. Padberg. Classification of Petri Nets Using Adjoint Functors *Bulletin of EACTS* 66, 1998.

[10] Note that these classes are already discussed regarding the a-posteriori semantics [7].

[11] For the difference of weak and strong capacities as well as the explicit synchronous semantics of weak capacities see [7].

[16] J. Padberg, H. Ehrig. Parametrized Net Classes: A uniform approach to net classes. LNCS 2128, pages 173–229, 2001.

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

[18] M.-O. Stehr, J. Meseguer and P. Ölveczky. Rewriting Logic as a Unifying Framework for Petri Nets. LNCS 2128, pages 250–303, 2001.

8. Appendix

In this Section we give detailed proofs of Lemma 11, Theorem 13, Lemma 16 and Lemma 22, which we omitted due to lack of space. For the proofs we need the following observations concerning the relationship between the $\sim$-axioms, the properties (Con1)-(Con6) and (Det), and the information associated to process terms: In 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 $\mathit{Inf}(\alpha) \subseteq/_{\cong} \mathit{Inf}(\beta)$. 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 order to sequentialize concurrently composed process terms we use axiom (4) with $\alpha_3 = \mathit{post}(\alpha_1)$ and $\alpha_2 = \mathit{pre}(\alpha_4)$. To transform concurrently composed step terms into synchronously composed step terms we additionally need axiom (7). 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 $(\alpha \,\|\, \beta) \sim (\alpha \,\|\, \mathit{pre}(\beta)); (\beta \,\|\, \mathit{post}(\alpha))$ and $\mathit{Inf}((\alpha \,\|\, \mathit{pre}(\beta)); (\beta \,\|\, \mathit{post}(\alpha))) \subseteq/_{\cong} \mathit{Inf}(\alpha \,\|\, \beta)$. If additionally (Con6) is fulfilled we get $\alpha \,\|\, \beta \sim \alpha \oplus \beta$ and $\mathit{Inf}(\alpha \oplus \beta) \subseteq/_{\cong} \mathit{Inf}(\alpha \,\|\, \beta))$.

These results will be used in equivalent transformations in the proofs of this Section mostly without mentioning them explicitly. If the associated information and sostructures 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 equivalent transformations. Finally we can conclude with axiom (10) for (Det1) and additionally axioms (9), (4), (8) and (7) for (Det2):

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

(Det2) If $(\mathcal{A}, \mathit{inf})$ fulfills (Det), (Con1) and (Con3)-(Con5), then there holds for $s_1, s_2 \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ and $m, m_1, m_2 \in M$: $(s_1 \oplus s_2) \,\|\, m$ and $(s_1 \,\|\, m_1); (s_2 \,\|\, m_2)$ defined with $\mathit{pre}((s_1 \oplus s_2) \,\|\, m) = \mathit{pre}((s_1 \,\|\, m_1); (s_2 \,\|\, m_2)) \Longrightarrow$

$post((s_1 \oplus s_2) \parallel m) \quad = \quad post((s_1 \parallel m_1); (s_2 \parallel m_2))$ and $(s_1 \oplus s_2) \parallel m \sim (s_1 \parallel m_1); (s_2 \parallel m_2)$.

We need the following technical notations concerning copy-nets.

In order to formulate basic relations between process terms of $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ and of $(\mathcal{A}, \mathit{inf})$, we extend the labelling function l from the definition of copy nets 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 $(\mathcal{A}, \mathit{inf})$: $l(\alpha) \in \mathcal{P}(\mathcal{A}, \mathit{inf})$, $pre(\alpha) = pre(l(\alpha))$, $post(\alpha) = post(l(\alpha))$, $\mathit{Inf}(\alpha) = \mathit{Inf}(l(\alpha))$, $\alpha \sim \beta \Longrightarrow l(\alpha) \sim l(\beta)$, $l : \mathcal{P}(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)}) \to \mathcal{P}(\mathcal{A}, \mathit{inf})$ is surjective, $go = (V, \prec, \sqsubset, id)$ associated to $\alpha \Longrightarrow go' = (V, \prec, \sqsubset, l)$ associated to $l(\alpha)$, $go' = (V, \prec, \sqsubset, l)$ associated to α' $\Longrightarrow \exists \alpha$ with $l(\alpha) = \alpha'$ such that $go = (V, \prec, \sqsubset, id)$ is associated to α, $go = (V, \prec, \sqsubset, id)$ and $go' = (V, \prec', \sqsubset', id)$ associated to $\alpha \Longrightarrow go = go'$.

Let $go = (V, \prec, \sqsubset, l)$ be an so-structure of a process term α of $(\mathcal{A}, \mathit{inf})$. A *copy term of α (w.r.t. go)* is a process term α^{go} of the (V, l)-copy net $(\mathcal{A}_{(V,l)}, \mathit{inf}_{(V,l)})$ with $l(\alpha^{go}) = \alpha$ such that $(V, \prec, \sqsubset, id)$ is associated to α^{go}. Two copy terms of α w.r.t. go are always $\sim$-equivalent through commutativity and associativity axioms. We will not further distinguish such copy terms.

Proof of Lemma 11. We proof 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 $\alpha \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ 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 $\alpha \in \mathit{Step}_{(\mathcal{A},\mathit{inf})}$ 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 8 there exist disjoint sets $V_\beta \subseteq V$ and $V_\gamma \subseteq V$ with $V_\beta \cup V_\gamma = V$, such that $go_\beta = (V_\beta, \prec \cap (V_\beta \times V_\beta), \sqsubset \cap (V_\beta \times V_\beta), id)$ is a labelled so-structure of β, $go_\gamma = (V_\gamma, \prec \cap (V_\gamma \times V_\gamma), \sqsubset \cap (V_\gamma \times V_\gamma), id)$ 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 $K_{go_\beta} \subseteq K_{go_\alpha}$ and $K_{go_\gamma} \subseteq K_{go_\alpha}$ are disjoint sets fulfilling $K_{go_\beta} \cup K_{go_\gamma} = K_{go_\alpha}$. Moreover, $k <_{go_\alpha} k'$ for each $k \in K_{go_\beta}$ and $k' \in K_{go_\gamma}$. Because α has no sub-term of the form $m; \alpha'$ we have $V_\beta \neq \emptyset$ and thus $B_\beta \neq \emptyset$ and $K_{go_\beta} \neq \emptyset$. 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 $\beta_C = ((\parallel_{s \in C} s) \parallel m_C); \beta_{su}$ is a defined process term fulfilling (I) and (II). Furthermore we have $\gamma_{su} = \gamma$, because B_γ and C are disjoint. Consequently $\alpha_C = ((\parallel_{s \in C} s) \parallel m_C); \alpha_{su} = ((\parallel_{s \in C} s) \parallel m_C); \beta_{su}; \gamma_{su} = \beta_C; \gamma$ is a defined process term (because $post(\beta_C) = post(\beta) = pre(\gamma)$). From $\beta_C \sim \beta$ we get (I) and from $\mathit{Inf}(\beta_C) \subseteq /_{\cong} \mathit{Inf}(\beta)$ 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 $go_\beta = (V_\beta, \prec \cap (V_\beta \times V_\beta), \sqsubset \cap (V_\beta \times V_\beta), id)$ is a labelled so-structure of β, $go_\gamma = (V_\gamma, \prec \cap (V_\gamma \times V_\gamma), \sqsubset \cap (V_\gamma \times V_\gamma), id)$ 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 $B_\beta \cup B_\gamma = B_\alpha$, $K_{go_\beta} \subseteq K_{go_\alpha}$ and $K_{go_\gamma} \subseteq K_{go_\alpha}$ are disjoint sets fulfilling $K_{go_\beta} \cup K_{go_\gamma} = K_{go_\alpha}$ and $k \not<_{go_\alpha} k'$ and $k' \not<_{go_\alpha} k$ for each $k \in K_{go_\beta}$, $k' \in K_{go_\gamma}$. 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 $\beta_{C_1} = ((\parallel_{s \in C_1} s) \parallel m_1); \beta_{su}$ and $\gamma_{C_2} = ((\parallel_{s \in C_2} s) \parallel m_2); \gamma_{su}$ are defined process terms fulfilling (I) and (II). From $\mathit{Inf}(\beta_{C_1}) \subseteq /_{\cong} \mathit{Inf}(\beta)$ and $\mathit{Inf}(\gamma_{C_2}) \subseteq /_{\cong} \mathit{Inf}(\gamma)$ we derive that $\beta_{C_1} \parallel \gamma_{C_2}$ is defined with $\mathit{Inf}(\beta_{C_1} \parallel \gamma_{C_2}) \subseteq /_{\cong} \mathit{Inf}(\beta \parallel \gamma) = \mathit{Inf}(\alpha)$. 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: $\beta_{C_1} \parallel \gamma_{C_2} = (((\parallel_{s \in C_1} s) \parallel m_1); \beta_{su}) \parallel (((\parallel_{s \in C_2} s) \parallel m_2); \gamma_{su}) \sim (((\parallel_{s \in C_1} s) \parallel m_1) \parallel ((\parallel_{s \in C_2} s) \parallel m_2)); (\beta_{su} \parallel \gamma_{su}) \sim ((\parallel_{s \in C} s) \parallel m_C); \alpha_{su} = \alpha_C$. 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 $\mathit{Inf}(\alpha_C) \subseteq /_{\cong} \mathit{Inf}(\beta_{C_1} \parallel \gamma_{C_2}) \subseteq /_{\cong} \mathit{Inf}(\beta \parallel \gamma) = \mathit{Inf}(\alpha)$. Consequently (II) is fulfilled. □

Proof of Theorem 13. Denote $go = go_\alpha = (V, \prec, \sqsubset, l)$ and $go' = (V, \prec', \sqsubset', l)$. We will equivalently transform α^{go} into the process term $\beta^{go'}$, i.e. $\beta^{go'}$ has associated the so-structure $(V, \prec', \sqsubset', id)$, then the process term $\beta = l(\beta^{go'})$ has associated the so-structure go' and satisfies $\alpha \sim \beta$. Consequently $pre(\beta) = pre(\alpha) = m$ and thus go is enabled in m.

Let $po_{go} = (K_{go}, <_{go})$ be the partial order associated to go, let $po_{go'} = (K_{go'}, <_{go'})$ be the total order associated to go' (see Remark 3). Let $K_{go'} = \{k'_1, \ldots, k'_m\}$ such that $k'_i <_{go'} k'_j \Leftrightarrow i < j$. Then k'_1 is of the form $k'_1 = k_1 \cup \ldots \cup k_n$ for (pairwise disjoint) $k_1, \ldots, k_n \in K_{go}$, where $k_1, \ldots, k_n$ are all minimal w.r.t. $<_{go}$ (see remark 3). Thus, the basic step terms $b_1, \ldots, b_n \in B_{\alpha^{go}}$ of α^{go} 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 α^{go} has no sub-term of the form $m; \alpha'$. According to corollary 12 there exists a marking m_{Sy} such that $\alpha^{go}_C = ((\oplus_{s \in C} s) \parallel m_{Sy}); \alpha^{go}_{su} = ((\oplus_{v \in k'_1} v) \parallel m_{Sy}); \alpha^{go}_{su}$ is a defined process term fulfilling (I) and (II). Thus we have detached k'_1 from α^{go}. Denote $V' = V \setminus \{v \mid \exists s \in C \text{ with } v \in s\}$, $K_C = K_{go} \setminus \{k_1, \ldots, k_n\}$ and $K'_C = K_{go'} \setminus \{k'_1\}$. Directly from the construction of α^{go}_{su} we get that $go_C = (V', \prec|_{V' \times V'}, \sqsubset|_{V' \times V'}, id)$ is an so-structure of α^{go}_{su} and $po_{go_C} = (K_C, <_{go}|_{K_C \times K_C})$ is the associated partial order.

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

Proof of Lemma 16. Denote $go = (V, \prec, \sqsubset, l)$, $go' = (V, \prec', \sqsubset', l)$ and $go'' = (V, \prec'', \sqsubset'', l)$. 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 go. As usual this is done on the copy term level. We start with the process term α. Without loss of generality $\alpha^{go'}$ is of the form $\alpha^{go'} = (s_1 \parallel m_1); \ldots; (s_n \parallel m_n)$ 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 go w.r.t. $\prec$. Without loss of generality each step term s_i is of the form $s_i = (s^{min}_i \oplus s^{rest}_i)$, where $s^{min}_i, s^{rest}_i \in Step_{(\mathcal{A}_{(V,l)}, inf_{(V,l)})}$ with $V_{min} \cap |s_i| = |s^{min}_i|$. Of course the terms s^{min}_i and s^{rest}_i are also allowed to be empty. In the following equivalence transformations s^{min}_i and s^{rest}_i 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 $(s^{min}_i \parallel m^{min}_i); (s^{rest}_i \parallel m^{rest}_i)$. Obviously for $v \in s^{min}_i, w \in s^{rest}_i$ we have $v \sqsubset' w$ and $w \sqsubset' v$. Because w is not minimal in go and v is minimal in go we conclude $w \not\sqsubset v$ (use (C4)). Consequently if we remove for each $i \in \{1, \ldots, n\}$ and each $v \in s^{min}_i, w \in s^{rest}_i$ the relation $w \sqsubset' v$ from go' and add in exchange the relation $v \prec' w$ to go' the result is a total linear so-structure $go'_1 = (V, \prec'_1, \sqsubset'_1, l)$ which extends go. Since go is enabled, there exists a synchronous step sequence term α'_1 with $pre(\alpha'_1) = m$ and associated so-structure go'_1. Its copy term $(\alpha'_1)^{go'_1}$ has without loss of generality the form $(\alpha'_1)^{go'_1} = (s^{min}_1 \parallel m^{min}_1); (s^{rest}_1 \parallel m^{rest}_1) ; \ldots ; (s^{min}_n \parallel m^{min}_n) ; (s^{rest}_n \parallel m^{rest}_n)$ satisfying $(\alpha'_1)^{go'_1} \sim \alpha^{go'}$ (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 $v \in s^{min}_n, w \in s^{rest}_{n-1}$ that $w \not\sqsubset v$. Consequently if we remove for each $v \in s^{min}_n, w \in s^{rest}_{n-1}$ the relation $w \prec'_1 v$ from go'_1 and add in exchange the relation $v \sqsubset'_1 w$ to go'_1 the result is a total linear so-structure $go'_2 = (V, \prec'_2, \sqsubset'_2, l)$ which extends go. As above there is a synchronous step sequence term α'_2 with associated so-structure go'_2 and the copy term of α'_2 w.r.t. go'_2 has without loss of generality the form $(s^{min}_1 \parallel m^{min_1}_1); (s^{rest}_1 \parallel m^{rest_1}_1) ; \ldots ; ((s^{min}_n \oplus s^{rest}_{n-1}) \parallel m^{minrest_1}_{n-1}); (s^{rest}_n \parallel m^{rest_1}_n)$ being equivalent to $(\alpha'_1)^{go'_1}$ (using (Det1) and (Det2)) and consequently to $\alpha^{go'}$. From go'_2 we now remove each relation $w \sqsubset'_2 v$ and add in exchange $v \prec'_2 w$ for $v \in s^{min}_n, w \in s^{rest}_{n-1}$ resulting in the total linear so-structure $go'_3 = (V, \prec'_3, \sqsubset'_3, l)$ which also extends go. With the same arguments we get an accordant copy term of the form $(s^{min}_1 \parallel m^{min_2}_1); (s^{rest}_1 \parallel m^{rest_2}_1); \ldots ; (s^{min}_n \parallel m^{min_2}_n) ; (s^{rest}_{n-1} \parallel m^{rest_2}_{n-1}); (s^{rest}_n \parallel m^{rest_2}_n)$ equivalent to $\alpha^{go'}$. Altogether the terms s^{min}_n and s^{rest}_{n-1} have been permuted. With a similar argumentation we can equivalently transform the sub-term $(s^{min}_{n-1} \parallel m^{min_2}_{n-1}); (s^{min}_n \parallel m^{min_2}_n)$ into $((s^{min}_{n-1} \oplus s^{min}_n) \parallel m^{min_3})$.

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 $(s^{min}_1 \oplus \ldots \oplus s^{min}_n \parallel m^{min'}); (s^{rest}_1 \parallel m^{rest'}_1) ; \ldots ; (s^{rest}_{n-1} \parallel m^{rest'}_{n-1}) ; (s^{rest}_n \parallel m^{rest'}_n) = ((\oplus_{v \in V_{min}} v) \parallel m^{min'}) ; (s^{rest}_1 \parallel m^{rest'}_1) ; \ldots ; (s^{rest}_{n-1} \parallel m^{rest'}_{n-1}) ; (s^{rest}_n \parallel m^{rest'}_n)$ equivalent to $\alpha^{go'}$ and with an associated so-structure go''' extending go. Thus, we have sorted the minimal events V_{min} of go to one synchronous step at the beginning of the term. Considering the so-structure $go_1 = (V \setminus V_{min}, \prec|_{(V \setminus V_{min}) \times (V \setminus V_{min})}, \sqsubset|_{(V \setminus V_{min}) \times (V \setminus V_{min})}, l)$ restricting go to the set of remaining events, we can collect in the same way the minimal events of go_1 to one synchronous step term at the second position of the synchronous step sequence term[12]. Now we re-iterate this procedure until go_i constructed in this way is empty. Because every go_i has minimal events the procedure terminates. Altogether we get a synchronous step sequence term $\alpha^{go'}_{go_{nat}}$ equiva-

[12]One must attention that the enabled property of go (and not of go_1) has to be used.

lent to $\alpha^{go'}$ that is uniquely defined by go up to concurrently composed markings and commutativity and associativity axioms[13]. The same can be applied to the process term β resulting in a copy term $\beta^{go''}_{go_{nat}}$ equivalent to $\beta^{go''}$. We deduce $\alpha^{go'}_{go_{nat}} \sim \beta^{go''}_{go_{nat}}$ from $pre(\beta^{go''}) = pre(\alpha^{go'})$, $\alpha^{go'}_{go_{nat}} \sim \alpha^{go'}$ and $\beta^{go''}_{go_{nat}} \sim \beta^{go''}$ (use (Det1)), so there results $\alpha^{go'} \sim \beta^{go''}$ and consequently $\alpha \sim \beta$. □

Proof of Lemma 22. It is enough to consider the cases where α is derived from β through one of the equivalent transformation axioms (1)-(11) (Definition 10). 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 $AON = (B, V, R, Act, l)$ (with $l|_V = id$) be the process represented by the run go.

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 $go_1 \subseteq go_2$ because in go_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 go_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 3): (a) $(t, c) \in R$ and $(c, s) \in R$ or (b) $(t, c) \in R$ and $(c, s) \in Act$ or (c) $(c, t) \in Act$ 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, 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. □

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