Theoretical Computer Science 262 (2001) 285–310
Theoretical Computer Science
www.elsevier.com/locate/tcs

Algebraic nets with flexible arcs ☆

Ekkart Kindler *, Hagen Völzer [1]

Humboldt-Universität zu Berlin, Institut für Informatik, D-10099 Berlin, Germany

Received February 1999; revised March 2000; accepted May 2000
Communicated by G. Rozenberg

Abstract

Algebraic Petri nets as defined by Reisig (Theoret. Comput. Sci. 80 (1991) 1–34.) lack a feature for modelling distributed network algorithms, viz. *flexible arcs*. In this paper, we equip algebraic Petri nets with flexible arcs and call the resulting extension *algebraic system nets*. We demonstrate that algebraic system nets are better suited for modelling distributed algorithms. Besides this practical motivation for introducing algebraic system nets, there is a theoretical one. The concept of *place invariants* introduced along with algebraic Petri nets has a slight insufficiency: There may be place invariants of the unfolded algebraic Petri net that cannot be expressed as a place invariant of the algebraic Petri net itself. By introducing algebraic system nets along with a more general concept of place invariants we eliminate this insufficiency too. Moreover, we generalize the concept of place invariants, which we call *simulations*. Many well-known concepts of Petri net theory such as *siphons*, *traps*, *modulo-invariants*, *sur-invariants* and *sub-invariants* are special cases of a simulation. Still, a simulation can be verified in the same style as classical place invariants of algebraic Petri nets.

Keywords: Algebraic Petri nets; Flexible arcs; Linear-algebraic verification techniques; Place invariants

0. Introduction

Algebraic Petri nets as proposed by Reisig [21] lack a feature that is convenient for modelling distributed network algorithms: Arcs with flexible throughput – *flexible arcs* for short – are not allowed. We will motivate the use and the necessity of flexible arcs with the help of an example. Then, we formally introduce a generalized version

☆ A preliminary version of this paper was published in [14].
* Corresponding author.
E-mail address: kindler@informatik.hu-berlin.de (E. Kindler).
[1] Supported by DFG: Konsensalgorithmen.

PII: S0304-3975(00)00205-X

of algebraic Petri nets that supports flexible arcs. We call this version *algebraic system nets*.

Algebraic system nets will be equipped with a concept of *place invariants*, which overcomes a problem of Reisig's version [21]. There, the unfolded algebraic Petri net may have a (low-level) place invariant that has no corresponding (high-level) place invariant in the algebraic Petri net. We will give an example for such a place invariant.

For convenience, we do not use the traditional representation of a place invariant as a vector of weight functions [10] or as a vector of terms [21]. Rather, we represent a place invariant as a *multiset-valued linear expression* in which place names may occur as bag-valued [2] variables. Though this difference is only syntactical, it allows a smoother transition between Petri net properties and temporal logic (cf. [22, 13, 27, 12]). Moreover, it gives rise to a generalization: We can use expressions that evaluate to an arbitrary *commutative monoid* equipped with some *affine preorder*. We call this generalization *simulation*. Algebraically, a simulation is a homomorphism from the occurrence graph of the net to the preordered commutative monoid. The use of linear weight functions into more general domains has been proposed before (cf. [25, 7]); the use of affine preorders, however, is new. It turns out that well-known concepts like *siphons* (*deadlocks*) and *traps* [19, 20], *modulo-invariants* [7], and *sur-invariants* and *sub-invariants* [16] are special cases of *simulations*. Traps and siphons for algebraic Petri nets have been already introduced by Schmidt [23]. Modulo-invariants, sub-invariants, and sur-invariants for algebraic nets are introduced in this paper as the canonical adaptation of the low-level versions. Moreover, we introduce *semi-place invariants* and *stabilization expressions* as further instances of simulations. Since all techniques are instances of the same concept, simulations allow us to apply these techniques in a uniform way. This simplifies the implementation of tools that support these verification techniques (see [1] for details).

The use of flexible arcs in algebraic Petri nets is not completely new. Billington [3, 4] proposed some extensions that allow a restricted kind of 'flexibility', and Reisig [21] indicated some possible extensions. Our definition of algebraic system nets has been introduced in [11] – without any results and without the concept of place invariants. Here, we present the above-mentioned results about algebraic system nets and the definition and investigation of place invariants. The relation of *algebraic system nets* with the versions of *algebraic Petri nets* of Vautherin [26] and Reisig [21] will be discussed in the conclusion.

In this paper, we define the semantics of an algebraic system net in two ways: We define the *processes* of an algebraic system net as a behavioural semantics, and we define the *unfolding* of an algebraic system net to a place/transition system. Unfoldings will be used to relate the concept of a place invariant of an algebraic system net to the classical concept of a place invariant of a place/transition-system. Since place/transition-systems have a behavioural semantics of their own, we have two

[2] In our terminology, a bag is a finite non-negative multiset.

behavioural semantics for algebraic system nets: the processes of the algebraic system net itself and the processes of its unfolding. We show that both concepts coincide.

The paper is organized as follows. In Section 1, we informally introduce algebraic system nets and motivate the need for flexible arcs. Moreover, we informally introduce our notation for place invariants and the generalization to simulations. Then, we formally define algebraic system nets and their processes in Section 2 and their place invariants in Section 3. In Section 4, we define unfoldings of algebraic system nets and discuss the relation of place invariants of an unfolding to the place invariants of the algebraic system net itself. The generalization of place invariants to simulations will be defined in Section 5. Last, we show that the processes of the unfolding are identical to the processes of the algebraic system net itself.

1. An example

Before we formally introduce *algebraic system nets*, we present an example, which models a simple distributed algorithm. The example motivates the need for flexible arcs and provides some intuitive understanding of algebraic system nets and the concept of place invariants.

1.1. A minimum distance algorithm

The algorithm works on a network of *agents* where some distinguished agents are so-called *roots* of the network. The algorithm computes the minimal distance from a root, for each agent of the network. This algorithm was inspired by a simple spanning tree algorithm [6]; the net model was presented already in [11] and verified in [12].

We denote the set of agents by A, the set of distinguished root-agents by $R \subseteq A$; the set of other agents, the set of the so-called *inner agents*, is denoted by $I = A \backslash R$. The underlying network is denoted by $N \subseteq A \times A$. The algebraic system net Σ_1 shown in Fig. 1 models the behaviour of each agent $x \in A$: Initially, a root-agent $x \in R$ sends a message to each of its *neighbours* in the network. With this message, it informs its neighbours that they have distance 1 from a root (viz. from x itself). The agent $x \in R$ makes an entry for itself that its distance from a root is 0. The currently known distance n of an agent x from some root agent is represented as a pair (x, n) on place distance. So, an agent may be in exactly one of the three states rootagent, inneragent or it knows some distance from a root. The behaviour of a root agent is modelled by *transition* t1 of Σ_1; a message m to an agent $y \in A$ is represented as a pair (y, m) on place messages. Suppose $y_1, \ldots, y_n$ are the neighbours of x in the communication network, then $M(x, 1)$ denotes the set of pairs [3] $[(y_1, 1), \ldots, (y_n, 1)]$, where each pair represents a message to one neighbour.

An *inner agent* $x \in I$ waits until it receives a message from some of its neighbours. When it receives a message, it accepts the distance n from this message; in addition,

[3] The use of square brackets indicates that we actually use bags rather than sets.

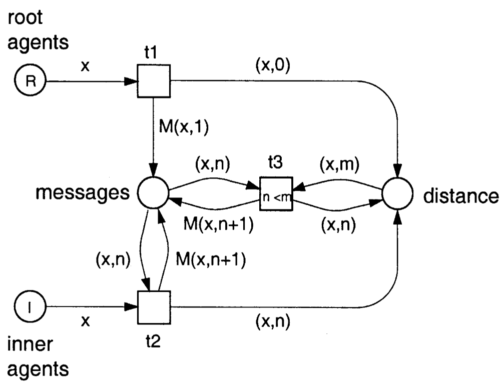

Fig. 1. A minimum distance algorithm Σ_1.

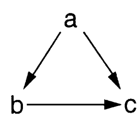

Fig. 2. A network of agents.

it sends a message $n+1$ to each of its neighbours. This behaviour is modelled by transition t2.

When an agent $x \in A$ receives another message with a distance n that is shorter than the distance m which it already knows, it accepts the new distance n and sends the new distance $n+1$ to each of its neighbours. This behaviour is modelled by transition t3, where the transition guard guarantees $n < m$. Altogether, this behaviour guarantees that eventually each agent knows its minimal distance to a root – if there is a path to some root at all.

Let us consider how messages are sent out in Σ_1 in more detail: As we said above, a message to an agent x is modelled as a pair (x, n) on place messages where n represents the contents of the message – in our example a number. In order to get a simple and concise Petri net model of the algorithm, we have modelled the sending of messages to all neighbours by a single transition; this is possible because $M(x, 1)$ resp. $M(x, n)$ represents a set of messages. Of course, the set denoted by $M(x, n)$ depends on the agent x and the underlying network N. For the network shown in Fig. 2, we have: $M(a, n) = [(b, n), (c, n)]$, $M(b, n) = [(c, n)]$, and $M(c, n) = []$ for each $n \in \mathbb{N}$, where $[]$ denotes the empty bag. For this network, the number of pairs in $M(x, n)$ varies for the different agents. Therefore, the number of tokens 'flowing through' the arc from transition t1 to place messages varies between 0 and 2. This is a typical example for a flexible arc. Therefore, Σ_1 is not a conventional algebraic Petri net as defined by Reisig [21].

Of course, it is possible to model the above algorithm by a conventional algebraic Petri net. For example, one could send the messages to each neighbour one after the other. But, the resulting algebraic Petri net has more transitions and is more complicated than Σ_1; the simplicity of Σ_1 results from the use of flexible arcs. Moreover, sending messages to each neighbour in some order is a design decision, which is completely irrelevant for the correctness of the algorithm. In this sense, the above model represents the algorithmic idea more concisely.

1.2. Place invariants as linear expressions

In our setting, a *place invariant* of an algebraic system net is represented by a linear expression in which place names of the net may occur as variables (of the corresponding bag type). Such an expression is, for example, $\mathsf{rootagents} + \mathsf{inneragents} + pr_1(\mathsf{distance})$. The function $pr_1 : A \times \mathbb{N} \to A$ is the projection of pairs to the first component. In order to apply this function to the bag distance, we linearly extended it to a function $pr_1 : \mathbf{B}(A \times \mathbb{N}) \to \mathbf{B}(A)$, where $\mathbf{B}(A)$ denotes the bags over the set A.

Given a marking, the expression evaluates to some multiset. Each place name stands for the bag of tokens at that place at the given marking. The example expression evaluates to the multiset[4] $R + I + [] = A$ in the initial marking. A linear expression is a place invariant if for each occurrence of a transition the expression evaluates to the same value at the marking before and at the marking after this occurrence.

The expression $\mathsf{rootagents} + \mathsf{inneragents} + pr_1(\mathsf{distance})$ is a place invariant of the above algebraic system net Σ_1. Since this expression evaluates to A in the initial marking, we can conclude that in each reachable marking of the system, the proposition $\mathsf{rootagents} + \mathsf{inneragents} + pr_1(\mathsf{distance}) = A$ holds. This property implies the previously mentioned observation that each agent is in exactly one of the three states rootagent, inneragent or distance. Representing a place invariant by an expression rather than by a vector of weights allows an easy integration of place invariants into a temporal logic framework: For a place invariant u that evaluates to a multiset A in the initial marking, we derive the temporal formula $u = A$.

To verify that a linear expression is a place invariant of the system, we have to check the validity of one equation for each transition. We consider transition t1 as an example. We construct the equation as follows: For the left-hand side of the equation we take the expression $\mathsf{rootagents} + \mathsf{inneragents} + pr_1(\mathsf{distance})$ and substitute each place name by the inscription of the arc from that place to transition t1, and we substitute [], when no arc exists. This gives us $x + [] + pr_1([])$. For the right-hand side we substitute each place name by the inscription of the arc from t1 to that place; this gives us $[] + [] + pr_1((x, 0))$. Obviously, the resulting equation $x + [] + pr_1([]) = [] + [] + pr_1((x, 0))$ is valid.

The substitutions for the left-hand side and the right-hand side of the equation corresponding to a transition t will be denoted by t^- and t^+, respectively. Then, a linear expression u is a place invariant of the algebraic system net, if for each transition t

[4] We treat sets as multisets by identifying them with their characteristic function.

of the algebraic system net the equation $t^-(u) = t^+(u)$ holds true (in the underlying algebra).

Usually, place invariants are characterized as follows: For each transition, $t^+ - t^-$ constitutes one column of the transposed incidence matrix N^T of the algebraic Petri net [21]. Then, a place invariant is a vector i of multiset-valued terms satisfying $N^T \cdot i = \underline{0}$, where multiplication is term substitution. Our syntactical representation is just a different view, which is more convenient for correctness proofs because it allows a smoother transition from place invariant equations to temporal propositions (cf. [13]). This, however, is only a matter of taste. What makes our concept of place invariants more powerful is that we also allow 'flexible expressions' in place invariants, which will be demonstrated in Section 4. Note that this would also be possible in vector notation.

1.3. More linear expressions

A place invariant is a linear expression of some multiset type. Its verification condition for each transition t is $t^-(u) = t^+(u)$. Now, let u be a linear expression of some monoid type X, and let $\hookrightarrow \subseteq X \times X$ be an affine[5] preorder in the monoid. Then, we say that $\hookrightarrow$ *simulates* Σ *via* u if $t^-(u) \hookrightarrow t^+(u)$. If u evaluates to u_0 in the initial marking, we have $u_0 \hookrightarrow u$ for each reachable marking.

From $u_0 \hookrightarrow u$, we can infer invariance properties of Σ. For example, if we choose the monoid $(2^A, \cup, \emptyset)$ and the preorder $\supseteq$, then, Σ_1 is simulated via the linear expression $supp(\mathsf{rootagents}) \cup supp(\mathsf{inneragents})$, where $supp$ denotes the *support* of a bag, i.e. the set of elements which occur at least once in the bag. We can conclude that for each reachable marking of Σ_1 holds $A \supseteq supp(\mathsf{rootagents}) \cup supp(\mathsf{inneragents})$.

Such an expression is called (*individual*) *siphon* of Σ: A transition adds a particular token to the siphon only if that token is also removed by that transition. Other verification techniques such as *traps* and *modulo-place invariants* can be formalized similarly. Moreover, we introduce *semi-place invariants* and *stabilization expressions* as further useful instances of simulations.

If an algebraic system net is simulated by a well-founded affine order, then the corresponding expression is called *stabilization expression*. Transitions which strictly decrease the value of the stabilization expression can happen only finitely many times. A special case of stabilization is termination: A *termination expression* proves that each run is finite. Sometimes, in Petri net theory, *sur-place invariants* and *sub-place invariants* [16] are used to prove termination. They are closely related to termination expressions and they will also be defined as special simulations.

As all these verification techniques are instances of the same scheme, they can be checked in the same way, by the simple local condition $t^-(u) \hookrightarrow t^+(u)$. The unification of verification techniques is one of the main benefits of our approach.

[5] A relation $\hookrightarrow$ is affine if, for each $x \hookrightarrow y$ and each z, we have also $z + x \hookrightarrow z + y$.

2. Algebraic system nets

In this section, we formalize algebraic system nets and their processes.

2.1. Basic notations

First, we introduce some notations and basic concepts from Petri nets [20] and algebraic specifications [8]. The only new concept is the *bag-signature* together with a corresponding concept of a *bag-algebra*.

Sets, families, functions, and relations: By $\mathbb{B}, \mathbb{N}$, and $\mathbb{Z}$ we denote the set {true, false} of truth values, the set of natural numbers with 0, and the set of integers, respectively. For a set A, we denote the cardinality of A by $|A|$, we denote the set of all non-empty finite sequences over A by A^+, and we denote the set of all subsets of A by 2^A. A *family* of sets over some *index set* I is denoted by $(A_i)_{i\in I}$. The family $(A_i)_{i\in I}$ is *pairwise disjoint*, if for each $i,j\in I$ with $i\neq j$ holds $A_i\cap A_j=\emptyset$. For a family $A=(A_i)_{i\in I}$, we use A also to denote the set $\bigcup_{i\in I} A_i$ when ambiguities are excluded by the context. For two sets A and B, we denote the set of all mappings from A to B by $B^A=\{f \mid f\colon A\to B\}$. If $R\subseteq A\times A$ is some relation over A then R^+ denotes the transitive closure of R, and R^* denotes the reflexive and transitive closure of R.

Monoids: A set A together with a commutative and associative binary operation $+$ and a neutral element 0 is called *commutative monoid*. A reflexive and transitive relation $\hookrightarrow\ \subseteq A\times A$ is *affine* if $\forall x,y,z\in A : x\hookrightarrow y\Rightarrow z+x\hookrightarrow z+y$. If $(A,+,0)$ is a commutative monoid and $\hookrightarrow\ \subseteq A\times A$ is an affine reflexive and transitive relation, $\mathscr{M}=(A,+,0,\hookrightarrow)$ is a *preordered commutative monoid*.

Let $\mathscr{M}=(A,+,0,\hookrightarrow)$ be a preordered commutative monoid and B be a set. By $\mathscr{L}_B(\mathscr{M})=(A^B,+_l,0_l,\hookrightarrow_l)$, we denote the *lifting* of $\mathscr{M}$ over B where $+_l, 0_l, \hookrightarrow_l$ are defined by $(f_1+_l f_2)(x)=f_1(x)+f_2(x)$, $0_l(x)=0$, and $f_1\hookrightarrow_l f_2$ if $\forall x\in B : f_1(x)\hookrightarrow f_2(x)$. We omit the index l where clear from the context. Obviously, $\mathscr{L}_B(\mathscr{M})$ is a preordered commutative monoid.

Multisets and bags: A *multiset* over a fixed set A is a mapping $M : A\to\mathbb{Z}$. The set of all multisets over A is denoted by $\mathbb{Z}^A$. We write $M[a]$ instead of $M(a)$ for the *multiplicity* of an element a in M. We define addition $+$, the empty multiset [], and inclusion $\leqslant$ of multisets by lifting $(\mathbb{Z},+,0,\leqslant)$ over A. The *support* of a multiset is defined by $supp(M)=\{x\in A \mid M[x]\neq 0\}$. A multiset M is *non-negative* if $M[x]\geqslant 0$ for all x in A, and M is *finite* if $supp(M)$ is finite. We consider sets as special multisets by identifying them with their characteristic function.

A finite non-negative multiset is also called *bag*. The set of all bags over A is denoted by $\mathbf{B}(A)$. We represent a bag by enumerating its elements in square brackets: $[a_1,\dots,a_n]$ (according to the multiplicities). We define the cardinality of a bag M by $|M|=\sum_{x\in A} M[x]$.

2.2. Place/transition systems

Petri nets: A *Petri net* (*net* for short) $N = (P, T, F)$ consists of two disjoint sets P and T and a relation $F \subseteq (P \times T) \cup (T \times P)$. An element of P is called *place*, an element of T is called *transition*, and an element of F is called *arc* of the net. As usual, we graphically represent a place by a circle, a transition by a square, and an arc by an arrow between the corresponding elements. A net is finite if both, P and T, are finite.

Place/transition systems: Basically, a place/transition system is a net with natural numbers as arc inscriptions. For convenience, we represent the arc inscriptions by two mappings $W^-, W^+ : T \to \mathbf{B}(P)$. The numbers $W^-(t)[p]$ and $W^+(t)[p]$ represent the inscription of arc (p, t) and (t, p), respectively. The respective number is 0 if and only if there is no such arc in the net.

Definition 1 (*Place/transition system*). A *place/transition system* $\Pi = (P, T, W^-, W^+, M_0)$ consists of
1. a set P of places and a set T of transitions, disjoint from P,
2. two mappings $W^-, W^+ : T \to \mathbf{B}(P)$,
3. a *marking* M_0, called *initial marking* of Σ, where a marking $M \in \mathbf{B}(P)$ of a place/transition system is a bag over P.

At a given marking $M_1 \in \mathbf{B}(P)$, a transition t is *enabled*, if there exists a marking M such that $M_1 = W^-(t) + M$, i.e. if $W^-(t) \leqslant M_1$. Then, transition t may *occur* resulting in the *successor marking* $M_2 = M + W^+(t)$. We denote the occurrence of transition t by $M_1 \xrightarrow{t} M_2$. If M_2 is a successor marking of M_1, we write $M_1 \to M_2$. If we have $M \xrightarrow{*} M'$, we say M' is *reachable* from M.

2.3. Algebras

Algebras and signatures: A *signature* $SIG = (S, OP)$ consists of a finite set S of *sort symbols* and a pairwise disjoint family $OP = (OP_a)_{a \in S^+}$ of *operation symbols*. A *SIG-algebra* $\mathscr{A} = ((A_s)_{s \in S}, (f_{op})_{op \in OP})$ consists of a family $A = (A_s)_{s \in S}$ of sets and a family $(f_{op})_{op \in OP}$ of total functions such that for each $op \in OP_{s_1 \dots s_n s_{n+1}}$ we have $f_{op} : A_{s_1} \times \cdots \times A_{s_n} \to A_{s_{n+1}}$. A set A_s of the algebra is called *domain* and a function f_{op} is called *operation* of the algebra.

Variables and terms: For a signature $SIG = (S, OP)$ we call a pairwise disjoint family $X = (X_s)_{s \in S}$ with $X \cap OP = \emptyset$ a *sorted SIG-variable set*. Next we define *terms*. Each term is associated with a particular sort. Let $X = (X_s)_{s \in S}$ be a sorted *SIG*-variable set. The *set of SIG-terms over X of sort s* is denoted by $\mathbf{T}_s^{SIG}(X)$ and inductively defined by:

1. $x \in X_s$ implies $x \in \mathbf{T}_s^{SIG}(X)$.

2. $u_i \in \mathbf{T}^{SIG}_{s_i}(X)$ for $i=1,\dots,n$ and $op \in OP_{s_1\dots s_n s_{n+1}}$ implies $op(u_1,\dots,u_n) \in \mathbf{T}^{SIG}_{s_{n+1}}(X)$.

The set of all terms (of any sort) is denoted by $\mathbf{T}^{SIG}(X)$. A term without variables is called *ground term*. We denote the set of ground terms by $\mathbf{T}^{SIG} = \mathbf{T}^{SIG}(\emptyset)$ and the set of ground terms of sort s by $\mathbf{T}^{SIG}_s = \mathbf{T}^{SIG}_s(\emptyset)$. We also write $\mathbf{T}_s$ for $\mathbf{T}^{SIG}_s(X)$ whenever SIG is clear from the context.

Evaluation of terms: For a signature $SIG = (S, OP)$, a sorted SIG-variable set $X = (X_s)_{s\in S}$, and a SIG-algebra $\mathscr{A} = ((A_s)_{s\in S}, (f_{op})_{op\in OP})$ a mapping $\beta : X \to A$ is an *assignment for X* if for each $s \in S$ and $x \in X_s$ holds $\beta(x) \in A_s$. We inductively extend β to a mapping $\beta : \mathbf{T}(X) \to A$ by

$$\beta(op(u_1,\dots,u_n)) = f_{op}(\beta(u_1),\dots,\beta(u_n)) \text{ for } op(u_1,\dots,u_n) \in \mathbf{T}(X).$$

The mapping β is called *β-evaluation in $\mathscr{A}$*. Let $\beta_\emptyset : \emptyset \to A$ be the unique assignment for the empty variable set; $\beta_\emptyset$ evaluates ground terms.

Substitutions: Let X and Y be SIG-variable sets. A mapping $\sigma : X \to \mathbf{T}(Y)$ is called *substitution* if $x \in X_s$ implies $\sigma(x) \in \mathbf{T}_s(Y)$. Analogously to evaluations, we also extend σ to a mapping $\sigma : \mathbf{T}(X) \to \mathbf{T}(Y)$ in order to apply it to terms. In case of $Y = \emptyset$ we call σ *ground substitution*. For an assignment β and substitutions σ and τ, we write $\beta\sigma(t)$ short for $\beta(\sigma(t))$ and we write $\sigma\tau(t)$ short for $\sigma(\tau(t))$.

Bag-signatures and -algebras: We introduce bag-signatures as particular signatures. In a bag-signature we distinguish some *ground-sorts* and we assign a *bag-sort* to each ground-sort. In a bag-algebra the domain associated with a bag-sort must be the bags over the domain of the corresponding ground-sort.

Definition 2 (*Bag-signature*, *BSIG-algebra*). Let $SIG = (S, OP)$ be a signature and $GS, BS \subseteq S$; $BSIG = (S, OP, bs)$ is a *bag-signature* if $bs : GS \to BS$ is a bijective mapping. An element of GS is called *ground-sort*, an element of BS is called *bag-sort* of $BSIG$. A SIG-algebra $\mathscr{A}$ is a *BSIG-algebra* if for each $s \in GS$ holds $A_{bs(s)} = \mathbf{B}(A_s)$, i.e. if for each *ground-domain* the corresponding *bag-domain* is actually the set of all bags over the ground-domain.

In the following, we assume that a bag-signature $BSIG$ has a sort symbol $bool \in S$ and in each $BSIG$-algebra the corresponding domain is $A_{bool} = \mathbb{B}$. Furthermore, we assume that for each bag-sort the usual bag operations (e.g. $\cdot + \cdot$, $[\cdot]$, $[]$) are predefined. A bag-signature $BSIG = (S, OP, bs)$ is a specialized signature $SIG = (S, OP)$ and by definition each $BSIG$-algebra is a SIG-algebra. Therefore, variables, terms, assignments, evaluation, and substitutions are defined for bag-signatures, too. For a boolean term $u \in \mathbf{T}_{bool}(X)$, we say that *u holds true* in $\mathscr{A}$ if for all assignments β of X, we have $\beta(u) = \text{true}$.

Sometimes, it will be necessary to extend an algebra of a net by some additional operations and sorts for analysis and verification purposes. Since the necessary extensions are not a priori known, it must be possible to extend an algebra without changing the original parts of the algebra. To this end, we define the concept of a *conservative extension* of an algebra. We call a signature $SIG = (S, OP)$ a *subsignature* of a signature $SIG' = (S', OP')$, if $S \subseteq S'$ and for each arity $a \in S^+$ we have $OP_a \subseteq OP'_a$. We also call SIG' an *extension* of SIG. Basically, SIG' may introduce new sorts and new operation symbols, but must not change the arity of operation symbols of SIG. A SIG'-Algebra $\mathscr{A}' = ((A'_s)_{s\in S}, (f'_{op})_{op\in OP'})$ is a *conservative extension* of a SIG-Algebra $\mathscr{A} = ((A_s)_{s\in S}, (f_{op})_{op\in OP})$ if for all $s \in S$ we have $A'_s = A_s$ and for all $op \in OP$ we have $f'_{op} = f_{op}$. This way, $\mathscr{A}'$ coincides with $\mathscr{A}$ for all sorts and operations of SIG.

Typically, we will extend a SIG-Algebra $\mathscr{A}$ by some sorts and operations of a specific structure. For example, we will use monoids, bags, or multisets. Technically, the extension $\mathscr{A}'$ is a conservative extension of two algebras: $\mathscr{A}$ and an algebra which represents the extended data type. For example, let $SIG = (S, OP)$ be some signature and $MSIG = (\{s\}, +_{sss}, 0_s)$ be a signature of a monoid (such that symbols s, $+$ and 0 do not occur in SIG). Then, we call $SIG' = (S \cup \{s\}, OP \cup \{+, 0\})$ a *monoid extension* of SIG. A SIG'-algebra $\mathscr{A}'$ is called a *monoid extension* of a SIG-Algebra $\mathscr{A}$ if it is a conservative extension of both, $\mathscr{A}$ and some monoid $\mathscr{M}$. Similarly, we refer to *bag* and *multiset extensions* if $\mathscr{M}$ is a bag or a multiset algebra.

2.4. Algebraic system nets

In this section, we define algebraic system nets, their markings, and the occurrence rule.

Definition 3 (*Algebraic system net*). Let $BSIG = (S, OP, bs)$ be a bag-signature with bag-sorts BS. An *algebraic system net* $\Sigma = (N, \mathscr{A}, X, i)$ over $BSIG$ consists of

1. a finite net $N = (P, T, F)$ where P is sorted over BS, i.e., $P = (P_s)_{s\in BS}$ is a bag-valued $BSIG$-variable set,
2. a $BSIG$-Algebra $\mathscr{A}$,
3. a sorted $BSIG$-variable set X disjoint from P,
4. a *net inscription* $i : P \cup T \cup F \to \mathbf{T}^{BSIG}(X)$ such that
   (a) for each $p \in P_s$: $i(p) \in \mathbf{T}^{BSIG}_s$, i.e., the restriction of i to P is a ground substitution for P,
   (b) for each $t \in T$: $i(t) \in \mathbf{T}^{BSIG}_{bool}(X)$, and
   (c) for each $t \in T$, and for each $p \in P_s$ with $f = (t, p) \in F$ or $f = (p, t) \in F$ holds $i(f) \in \mathbf{T}^{BSIG}_s(X)$.

For a place $p \in P$, the inscription $i(p)$ is called *symbolic initial marking* of p; for a transition $t \in T$, the term $i(t)$ is called *guard* of t. Note that a place is considered to be a variable and the sort of a place is a bag-sort.

Definition 4 (*Pre- and post-substitution*). For each transition t of an algebraic system net Σ we define the two substitutions $t^-, t^+ : P \to \mathbf{T}(X)$, called *pre- and post-substitution* respectively, by:

$$t^-(p) = \begin{cases} i(p,t) & \text{if } (p,t) \in F, \\ [] & \text{otherwise,} \end{cases} \qquad t^+(p) = \begin{cases} i(t,p) & \text{if } (t,p) \in F, \\ [] & \text{otherwise.} \end{cases}$$

Definition 3 gives the syntax of an algebraic system net. The algebra, however, is still given semantically because we want to be flexible. We can incorporate any appropriate formalism for representing an algebra. In the tool proposed in [1], for example, the algebra of an algebraic system net is characterized as a theory for a theorem prover for first-order predicate logic. The semantics, i.e. the processes, of an algebraic system net will be defined in Section 2.5. Here, we define *markings* and the *occurrence-rule* for algebraic system nets. A marking associates each place of an algebraic system net with a bag over the corresponding sort.

Definition 5 (*Marking and initial marking*). Let *BSIG* be a bag-signature and Σ be an algebraic system net as in Definition 3. A *marking* M *of* Σ is an assignment for P. The marking M_0 with $M_0(p) = \beta_\emptyset(i(p))$ for each $p \in P$ is called the *initial marking of* Σ. We define the addition and inclusion of markings by lifting bags over P.

Transitions of algebraic system nets occur in *modes*. A *mode* of a transition associates each variable of X with some value of the algebra. In a particular mode, an arc-inscription evaluates to some bag. A transition t may occur in mode μ if all elements denoted by the inscription of the arcs pointing to t are present in the current marking and the guard of the transition evaluates to true. We formalize the occurrence-rule by the help of the markings μt^- and μt^+. The marking μt^- and the marking μt^+ represent the elements which are removed and added, respectively, when t occurs in mode μ. A pair (t, μ) is also denoted by $t.\mu$. We call $t.\mu$ an *action*.

Definition 6 (*Occurrence rule and reachable markings*). Let Σ be an algebraic system net as in Definition 3. Let $t \in T$ and μ be an assignment for X in $\mathscr{A}$. In a given marking M_1, a transition t is enabled in mode μ if there exists a marking M such that $M_1 = \mu t^- + M$ and $\mu(i(t)) = \text{true}$. Then, transition t may occur in mode μ, which results in the *successor marking* $M_2 = M + \mu t^+$. We denote the occurrence of transition t in mode μ by $M_1 \xrightarrow{t.\mu} M_2$. If M_2 is any successor marking of M_1, then we denote this by $M_1 \longrightarrow M_2$. If we have $M \xrightarrow{*} M'$ then we say M' is reachable from M. We say that M is a *reachable marking of* Σ if M is reachable from M_0, i.e. the initial marking of Σ.

Remark 1. In the following, we only consider algebraic system nets in which for each transition t and each mode μ, the markings μt^- and μt^+ are nonempty. This helps to avoid some anomalies in the definition of processes (see [2] for further explanation).

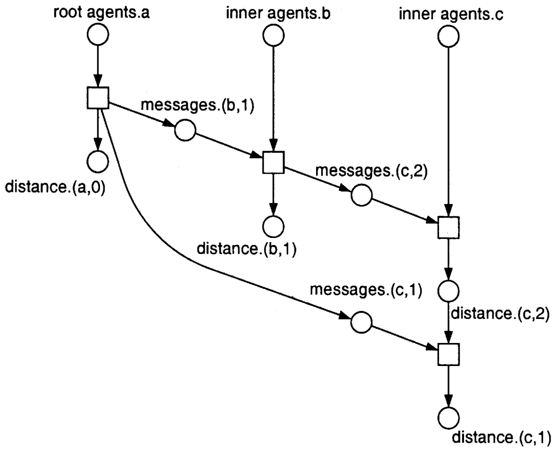

Fig. 3. A process of Σ_1.

2.5. *Processes of algebraic system nets*

Now, we define non-sequential processes [9, 2] for algebraic system nets. Fig. 3 shows a process of the algebraic system net Σ_1 (see Fig. 1) on the network shown in Fig. 2. Basically, a process is an inscribed acyclic Petri net with non-branching places. The inscription of the initial places corresponds to the initial marking of the algebraic system net and each transition corresponds to the occurrence of a transition of the algebraic system net in some mode.

For the formal definition of processes we start with some notations and definitions, which mainly follow the lines of [2].

Definition 7. Let $N = (P, T, F)$ be a net.

1. For an element $x \in P \cup T$ of N, we define the *preset* of x by ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ and the *postset* of x by $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$.
2. We define the *minimal elements of* N by ${}^\circ N = \{x \in P \cup T \mid {}^\bullet x = \emptyset\}$ and the *maximal elements of* N by $N^\circ = \{x \in P \cup T \mid x^\bullet = \emptyset\}$.
3. For $x \in P \cup T$ we define the *set of predecessors* by $\downarrow x = \{y \in P \cup T \mid (y, x) \in F^+\}$.

Processes are defined by the help of *occurrence nets*. An *occurrence net* has two main features: The flow relation is acyclic and is not branching at places. Moreover, each element of an occurrence net has only finitely many predecessors. For a detailed motivation of all features we refer to [9, 2].

Definition 8 (*Occurrence net*). A net $K = (B, E, \lessdot)$ is an *occurrence net* if

1. ${}^\circ K \subseteq B$ and $K^\circ \subseteq B$,

2. $^\circ K$ is finite and for each $e \in E$ both, ${}^\bullet e$ and $e^\bullet$, are finite,
3. for each $b \in B$ holds $|{}^\bullet b| \leqslant 1$ and $|b^\bullet| \leqslant 1$, and
4. for each $b \in B$ the set of predecessors $\downarrow b$ is finite and $b \notin \downarrow b$.

For the sake of clarity, we use other symbols for places and transitions in an occurrence net. Moreover, we call a place of an occurrence net *condition* and we call a transition *event*. Next we define the *states* of an occurrence net.

Definition 9 (*States of an occurrence net*). Let $K = (B, E, \lessdot)$ be an occurrence net. For two subsets of conditions $Q, Q' \subseteq B$ we define the occurrence relation $\longrightarrow$ by: $Q \longrightarrow Q'$ if there exists an event $e \in E$ such that ${}^\bullet e \subseteq Q$ and $Q' = (Q \setminus {}^\bullet e) \cup e^\bullet$. For $Q, Q' \subseteq B$ we say Q' *is reachable from* Q if $Q \stackrel{*}{\longrightarrow} Q'$. A subset of conditions $Q \subseteq B$ is a *state of* K, if Q is reachable from $^\circ K$. The set $^\circ K$ is called the *initial state* of K.

Processes of algebraic system nets: In a process, each condition of the occurrence net is associated with some place of the algebraic system net along with an element of the corresponding domain. This is formalized as condition labelling.

In the following, we use the notation $p.a$ for the pair (p, a) if p is a place and if a is a token of an algebraic system net.

Definition 10 (*Condition labelling*). Let Σ be an algebraic system net over a bag-signature *BSIG* as in Definition 3, and let $K = (B, E, \lessdot)$ be an occurrence net. A mapping $r : B \to P \times A$ is a *condition labelling of* K if for each $b \in B$ with $r(b) = p.a$ it holds that $a \in A_s$ implies $p \in P_{bs(s)}$. For a given condition labelling r, each finite subset $Q \subseteq B$ can be associated with a marking. We denote this marking by $r(Q)$ and define it by $r(Q) : P \to \mathbf{B}(A)$ with $r(Q)(p)[a] = |\{b \in Q \mid r(b) = p.a\}|$.

An occurrence net with labelled conditions is a *process* of an algebraic system net if the initial state is labelled by the initial marking and each event corresponds to the occurrence of a transition in some mode (cf. Fig. 3).

Definition 11 (*Process*). Let Σ be an algebraic system net, let $K = (B, E, \lessdot)$ be an occurrence net, and let r be a condition labelling of K. Then, (K, r) is a *process* of Σ, if

1. $r(^\circ K) = M_0$, where M_0 is the initial marking of Σ, and
2. for each event $e \in E$ there exists a transition $t \in T$ and a mode μ such that $\mu(i(t)) = \text{true}$, $r({}^\bullet e) = \mu t^-$, and $r(e^\bullet) = \mu t^+$.

Definition 11 is the canonical extension of processes [2] to algebraic system nets, which will be verified in Section 6.

3. Place invariants

In this section, we will define and investigate place invariants for algebraic system nets. As already shown in the introduction, we use a linear expression rather than a vector of terms to represent a place invariant. In this expression, a place is interpreted as a variable of the corresponding bag-sort.

Remark 2. In the following, we employ terms over mixed variable sets, for example a term $\varphi \in \mathbf{T}^{BSIG}_{bool}(P \cup Y)$. Since P and Y are assumed to be disjoint, an assignment M for P (i.e. a marking) and an assignment β for Y can be canonically composed to an assignment for $P \cup Y$, which is denoted by M^{β}.

Definition 12 (*Place invariant*). Let $BSIG$ be a bag-signature, $\Sigma = (N, \mathscr{A}, X, i)$ be an algebraic system net over $BSIG$ with places P and let $\mathscr{A}'$ be a SIG'-algebra that is a multiset extension of $\mathscr{A}$ with multiset sort s. Furthermore, let Y be a variable set for SIG' disjoint from P and let $v \in \mathbf{T}^{SIG'}_{s}(Y \cup P)$ be a multiset-valued expression. An expression v is called a *place invariant* of Σ if and only if

1. v is linear, i.e. for each two markings M_1 and M_2 and each assignment β for Y, it holds that $(M_1 + M_2)^{\beta}(v) = M_1{}^{\beta}(v) + M_2{}^{\beta}(v)$, and
2. for all transitions t, the conditional equation $i(t) \Rightarrow t^{-}(v) = t^{+}(v)$ holds.

Remark 3. The integers can be considered to be multisets (multisets over $\{\bullet\}$). Therefore, integer-valued expressions are also place invariants if the corresponding conditions are satisfied.

Note that we defined linearity semantically. A syntactical characterization is straightforward and can be found in [27]. As already stated, the evaluation of a place invariant is constant for all reachable markings:

Proposition 1. *A linear multiset-valued expression $v \in \mathbf{T}_s(Y \cup P)$ is a place invariant of an algebraic system net Σ if and only if, for each transition t and each mode μ of Σ, each assignment β for Y, and each two markings M_1 and M_2 of Σ with $M_1 \xrightarrow{t.\mu} M_2$, we have $M_1{}^{\beta}(v) = M_2{}^{\beta}(v)$.*

This result is an immediate consequence of the definition of the occurrence rule and the definition of place invariants. We will formalize and prove a more general result in Theorem 2.

Reisig [21] represents a place invariant by a P-vector of multiset terms: A non-flexible multiset term is assigned to each place $p \in P$, which represents a function f_p. Here, non-flexibility means: For all markings M_1 and M_2 with $|M_1(p)| = |M_2(p)|$ we have $|f_p(M_1(p))| = |f_p(M_2(p))|$. An immediate consequence of this is the following: For f_p there exists a number n_p such that $|f_p(p)| = n_p \cdot |p|$. The vector notation of [21] translates to the linear expression $f_{p_1}(p_1) + f_{p_2}(p_2) + \cdots + f_{p_n}(p_n)$.

4. Unfoldings

In Section 2.5, we have defined the semantics of an algebraic system net in terms of processes. An alternative approach is to define the semantics of an algebraic system net by *unfolding* it to a place/transition system (e.g. [24]). Here, we will define the unfolding of an algebraic system net. The main reason, however, for defining unfoldings is that we want to relate the place invariants of an algebraic system net with the place invariants of its unfolding.

In this section, we first present the definition of an unfolding. Then, we give an example of an algebraic Petri net [21] which has a place invariant in the unfolding but no corresponding place invariant (according to the definition of [21]) in the algebraic Petri net itself. Last we will show, that this does no longer hold for our version of place invariants: According to our definition each place invariant of the unfolding has a corresponding place invariant in the algebraic system net itself.

4.1. Definition of the unfolding

The unfolding of an algebraic system net is a place/transition-system. The main idea of the unfolding is the following: Each transition of the unfolding corresponds to a transition of the algebraic system net in a particular mode; i.e. an action. Each place corresponds to a place of the algebraic system net projected to a particular element on that place. Technically, a transition of the unfolding is a pair of a transition t of the algebraic system net and a mode μ; a place of the unfolding is a pair of a place of the algebraic system net and an element a of the corresponding domain. Arcs and the arc-inscriptions transfer accordingly.

Definition 13 (*Unfolding*). Let $\Sigma=(N,\mathscr{A},X,i)$ be an algebraic system net over $BSIG=(S,OP,bs)$ with $N=(P,T,F)$, ground sorts GS, and initial marking M_0. We define

1. $\widehat{P}=\{p.a \mid s\in GS, p\in P_{bs(s)}, a\in A_s\}$
2. $\widehat{T}=\{t.\mu \mid t\in T, \mu$ is an assignment for X with $\mu(i(t))=\text{true}\}$
3. $W^-:\widehat{T}\to\mathbf{B}(\widehat{P})$ by $W^-(t.\mu)[p.a]=\mu t^-(p)[a]$ for $t.\mu\in\widehat{T}$ and $p.a\in\widehat{P}$.
4. $W^+:\widehat{T}\to\mathbf{B}(\widehat{P})$ by $W^+(t.\mu)[p.a]=\mu t^+(p)[a]$ for $t.\mu\in\widehat{T}$ and $p.a\in\widehat{P}$.
5. $\widehat{M_0}\in\mathbf{B}(\widehat{P})$ by $\widehat{M_0}[p.a]=M_0(p)[a]$ for $p.a\in\widehat{P}$.

The place/transition system $\hat{\Sigma}=(\hat{P},\hat{T},W^-,W^+,\widehat{M_0})$ is called the *unfolding of* Σ.

An example of an algebraic system net Σ_2 and its unfolding $\widehat{\Sigma_2}$ can be found in Figs. 4 and 5, where we assume that the domain of both places is $\mathbf{B}(\{a,b\})$. This is a very simple example since each transition has exactly one mode (there are no variables).

For a marking M of an algebraic system net Σ, we denote the corresponding marking in the unfolding $\widehat{\Sigma}$ by $\widehat{M}$, where $\widehat{M}$ is defined by $\widehat{M}[p.a]=M(p)[a]$ for each $p.a\in\widehat{P}$. Obviously, the occurrence rule of an algebraic system net and its unfolding coincide in the following way:

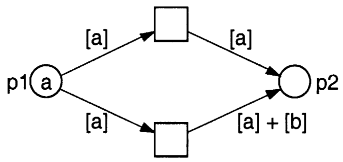

Fig. 4. An algebraic system net Σ_2.

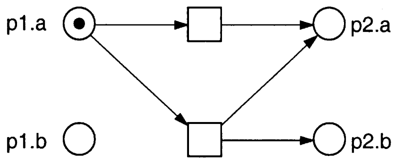

Fig. 5. The unfolding $\widehat{\Sigma_2}$.

Proposition 2. *Let Σ be an algebraic system net, let M_1 and M_2 be markings of Σ, and let $t.\mu$ be an action of Σ. Then, we have $M_1 \xrightarrow{t.\mu} M_2$ if and only if $t.\mu$ is a transition of $\widehat{\Sigma}$ and $\widehat{M_1} \xrightarrow{t.\mu} \widehat{M_2}$.*

In Section 6, we will generalize this result by proving that the processes of the unfolding coincide with the processes of the algebraic system net itself.

4.2. Place invariants of place/transition systems

Before defining the correspondence between place invariants of algebraic system nets and their unfoldings, let us briefly rephrase the concept of place invariants for our definition of place/transition systems. A *place invariant* of a place/transition system associates a weight with each place such that the weighted sum of tokens is the same for all reachable markings.

Definition 14 (*Place invariants of place/transition systems*). Let $\Pi = (P, T, W^-, W^+, M_0)$ be a place/transition system. A weight function $j : P \to \mathbb{Z}$ is called a *place invariant* of Π, if for each transition $t \in T$ the equation [6]

$$\sum_{p \in P} j(p) \cdot W^-(t)[p] = \sum_{p \in P} j(p) \cdot W^-(t)[p]$$

holds.

Similar to Proposition 1, there is a behavioural characterization of place invariants for place/transition systems.

[6] This equation is often written in vector notation: $j \cdot W^- = j \cdot W^+$.

Proposition 3. *Let Π be a place/transition system. A weight function $j: P \to \mathbb{Z}$ is a place invariant of Π, if and only if for each transition t and each two markings M_1 and M_2 of Π with $M_1 \xrightarrow{t} M_2$ we have $\sum_{p \in P} j(p) \cdot M_1[p] = \sum_{p \in P} j(p) \cdot M_2[p]$.*

In the following, we will write $j \cdot M$ as a shorthand for $\sum_{p \in P} j(p) \cdot M[p]$.

4.3. Correspondence of place invariants

In this section, we will show that there is an exact correspondence between the place invariants of an algebraic system net and its unfolding. Before formalizing this result, let us give a counter-example which shows that in the formalism of Reisig [21] there exists a place invariant of the unfolding which has no corresponding place invariant in the algebraic system net itself: Consider the algebraic Petri net Σ_2 of Fig. 4, where a and b are two different constants of the same sort. Fig. 5 shows the unfolding $\widehat{\Sigma_2}$ of this algebraic system net. Obviously, $j = \mathsf{p1}.a + \mathsf{p2}.a$ is a place invariant of $\widehat{\Sigma_2}$. Now, we will show that Σ_2 has no place invariant that corresponds to j, when we restrict ourselves to non-flexible expressions. Actually, we show that Σ_2 has only a trivial non-flexible place invariant. Assume that a non-flexible expression u is a place invariant of Σ_2. Then, u can be represented by $f_1(\mathsf{p1}) + f_2(\mathsf{p2})$. It follows that $|f_1(\mathsf{p1}) + f_2(\mathsf{p2})|$ is also a place invariant of Σ_2 which can equivalently be rewritten to $|f_1(\mathsf{p1})| + |f_2(\mathsf{p2})|$. Since the invariant u is non-flexible by assumption, we know that there exist integer values n_1 and n_2 such that $|f_1(\mathsf{p1})| + |f_2(\mathsf{p2})| = n_1 \cdot |\mathsf{p1}| + n_2 \cdot |\mathsf{p2}|$. By definition, this expression is a place invariant if and only if the following two equations hold true: $n_1 \cdot |[a]| + n_2 \cdot |[]| = n_1 \cdot |[]| + n_2 \cdot |[a]|$ and $n_1 \cdot |[a]| + n_2 \cdot |[]| = n_1 \cdot |[]| + n_2 \cdot |[a] + [b]|$. These equations can be simplified to $n_1 = n_2$ and $n_1 = 2 \cdot n_2$. This implies $n_1 = n_2 = 0$. Therefore, u is a trivial place invariant; i.e. u evaluates to the empty multiset [] for all markings. Since the only assumption imposed on u was that it is a non-flexible place invariant of Σ_2, we know that all non-flexible place invariants of Σ_2 are trivial. In particular, there is no non-flexible place invariant which corresponds to j.

The reason why there are only trivial place invariants of Σ_2 in the approach of Reisig [21] is that each token on a place is mapped to a multiset of the same cardinality. In order to express the invariant j of the unfolding, it is necessary to map a token a on places $\mathsf{p1}$ and $\mathsf{p2}$ to a singleton multiset (e.g. to $[a]$) and a token b to the empty multiset []. The invariant of $\widehat{\Sigma_2}$ from Fig. 5 can be formulated as a place invariant of Σ_2 by the expression $\mathsf{p1} + f_a(\mathsf{p2})$ where f_a is a linear function defined by $f_a([a]) = [a]$ and $f_a([b]) = []$, where f_a is not a legal function in the approach of [21].

In the above example, it was easy to show that there is no corresponding place invariant in the algebraic system net, because there was no non-trivial non-flexible place invariant at all. So, there was no need to formalize the concept of correspondence. In order to make the result precise, we need to formalize the concept of correspondence. The idea is quite simple: Each place invariant v canonically induces an equivalence $\equiv_v$ on the markings: Two markings are equivalent, if v evaluates to the same value in both markings. Furthermore, each marking M of an algebraic system net uniquely maps to a corresponding marking $\widehat{M}$ in the unfolding. Now, a place invariant v of an algebraic system net corresponds to a place invariant j of the unfolding Σ, if for all markings M_1 and M_2 of Σ we have $M_1 \equiv_v M_2$ if and only if $\widehat{M_1} \equiv_j \widehat{M_2}$. In the following formalization, we generalize this idea to sets of place invariants because a single place invariant of an algebraic system net in general corresponds to a set of place invariants of the unfolding.

Definition 15 (*Equivalence of markings*, *correspondence*).

1. Let Σ be an algebraic system net and V be a set of place invariants of Σ. The equivalence $\equiv_V$ on markings of Σ induced by V is defined by $M_1 \equiv_V M_2$ if and only if, for each $v \in V$ and each assignment β, we have $M_1{}^{\beta}(v) = M_2{}^{\beta}(v)$.
2. Let Π be a place/transition system and J be a set of place invariants of Π. The equivalence $\equiv_J$ on markings of Π induced by J is defined by $M_1 \equiv_J M_2$ if and only if, for each $j \in J$, we have $j \cdot M_1 = j \cdot M_2$.
3. Now, let $\widehat{\Sigma}$ be the unfolding of Σ. The set V of place invariants of Σ *corresponds* to the set of place invariants J of $\widehat{\Sigma}$ if and only if, for each two markings M_1 and M_2 of Σ, we have: $M_1 \equiv_V M_2$ if and only if $\widehat{M_1} \equiv_J \widehat{M_2}$.

In the following, we will show that, for each set V of place invariants of an algebraic system net Σ, there exists a corresponding set J of place invariants of the unfolding. Note that J may be infinite even for finite sets V. Vice versa, we show that, for each set J of place invariants of the unfolding, there exists a corresponding set of place invariants V of the algebraic system net. The second correspondence does not hold for Reisig's formalism [21].

Theorem 1. *Let Σ be an algebraic system net and let $\widehat{\Sigma}$ be its unfolding.*

1. *For each set of place invariants V of Σ, there exists a corresponding set of place invariants J of $\widehat{\Sigma}$.*
2. *For each set of place invariants J of $\widehat{\Sigma}$, there exists a corresponding set of place invariants V of Σ.*

Proof. 1. For each place $p \in P_{bs(s)}$ and each $a \in A_s$, let $M_{p.a}$ be the marking with $M_{p.a}(p)[a] = 1$ and $M_{p'.a'}(p')[a'] = 0$ for $p' \neq p$ or $a' \neq a$. For each place invariant $v \in \mathbf{T}_s^{SIG'}(Y \cup P)$ of Σ and each assignment β for Y, we define a mapping $j_v^{\beta} : \widehat{P} \to \mathbb{Z}$ by $j_v^{\beta}(p.a) = M_{p.a}^{\beta}(v)$.

Since v is linear, we have $M_{p.a}^{\beta}(v) = \sum_{s \in GS} \sum_{p \in P_{bs(s)}} \sum_{a \in A_s} M(p)[a] \cdot M_{p.a}^{\beta}(v) = \sum_{p \in \widehat{P}} \widehat{M}(p) \cdot j_v^{\beta}(p) = j_v^{\beta} \cdot \widehat{M}$. Thus, we have $M \equiv_v M'$ if and only if $\widehat{M} \equiv_{j_v^{\beta}} \widehat{M}'$ for all assignments β for Y. Therefore, a place invariant v of Σ corresponds to the set $\{ j_v^{\beta} \mid \beta \text{ is an assignment for } Y\}$.

By Proposition 2, we have for each two markings $\widehat{M_1}$ and $\widehat{M_2}$ and each $t.\mu \in \widehat{T}$ with $\widehat{M_1} \xrightarrow{t.\mu} \widehat{M_2}$: $M_1 \xrightarrow{t.\mu} M_2$. Since v is a place invariant of Σ, we know $M_1{}^{\beta}(v) = M_2{}^{\beta}(v)$ for each assignment β. Altogether, we have $j_v^{\beta} \cdot \widehat{M_1} = M_1{}^{\beta}(v) = M_2{}^{\beta}(v) = j_v^{\beta} \cdot \widehat{M_2}$.

By Proposition 3, j_v^β is a place invariant of $\widehat{\Sigma}$.
Now, let $J=\{j_v^\beta \mid v\in V, \beta$ is an assignment for $Y\}$. By definition and the above arguments, J is a set of place invariants corresponding to V.

2. For each place invariant j of $\widehat{\Sigma}$ and each place $p\in P_s$ of Σ, we extend the algebra by an operation $f_p^j : s\to\mathbb{Z}$ defined as the linear extension of $f_p^j([a])=j(p.a)$. Then, the expression $v=f_{p_1}^j(p_1)+\cdots+f_{p_n}^j(p_n)$ is linear. By definition, we have for each marking M of $\Sigma: M(v)=j\cdot\widehat{M}$. By the same arguments as above, we get that v is a place invariant of Σ which corresponds to j.

Altogether, the set $V=\{f_{p_1}^j(p_1)+\cdots+f_{p_n}^j(p_n)\mid j\in J\}$ is a set of place invariants of Σ which corresponds to J. □

5. More linear verification techniques

In this section, we show that, besides place invariants, there are other classical verification techniques that can be represented as linear expressions. Such techniques are traps, siphons, modulo-place invariants, sub- and sur-invariants. The benefit of this approach is twofold: On the one hand, we gain a common calculus for all these techniques, i.e. a common verification condition and common use of the techniques. On the other hand, other instantiations of linear expressions may lead to new verification techniques. This will be illustrated by introducing semi-place invariants and stabilization expressions.

We start with the central notion of this section, viz. simulations.

Definition 16 (*Linear expression*, *simulation*). Let *BSIG* be a bag-signature with sorts S and let $\Sigma=(N,\mathscr{A}, X,i)$ be an algebraic system net over *BSIG* with places P. Let $\mathscr{M}=(A_s,+,0,\hookrightarrow)$ be a preordered commutative monoid and let $\mathscr{A}'$ be a *SIG*-algebra such that $\mathscr{A}'$ is a conservative extension of $\mathscr{A}$ and $\mathscr{M}$. Let Y be a variable set for *SIG* disjoint from P.

A *Σ-expression* is a $\mathscr{M}$-valued term $u\in\mathbf{T}_s^{SIG'}(Y\cup P)$, which is called *linear* if:

$$\forall\beta:\ \forall M_1,M_2:\ (M_1+M_2)^\beta(\mu)=M_1^\beta(u)+M_2^\beta(u).$$

We say *$\mathscr{M}$ simulates Σ via u* if, for each transition t of Σ, the following condition is satisfied:

$$i(t)\Rightarrow t^-(u)\hookrightarrow t^+(u).$$

The following theorem is the basis for deriving invariance properties from simulations: The value of each reachable marking is related to the initial value (by $\hookrightarrow$), in other words: A marking, to which the expression is applied such that the resulting value does not relate to the initial value, is not reachable.

Theorem 2. *Let Σ be an algebraic system net with initial state M_0, and let $\mathscr{M}$ simulate Σ via u. Then, for each assignment β and for each reachable marking M of Σ we have ${M_0}^{\beta}(u) \hookrightarrow M^{\beta}(u)$.*

Proof. Let β be an arbitrary assignment. First we show that $M_1 \xrightarrow{t.\mu} M_2$ implies ${M_1}^{\beta}(u) \hookrightarrow {M_2}^{\beta}(u)$ for all markings M_1, M_2 of Σ: If we have $M_1 \xrightarrow{t.\mu} M_2$ then we have $\mu(i(t)) = \text{true}$ and it exists a marking M such that $M_1 = \mu t^- + M$ and $M_2 = M + \mu t^+$. Since $\mathscr{M}$ simulates Σ via u, we get $(\mu t^-)^{\beta}(u) \hookrightarrow (\mu t^+)^{\beta}(u)$. By affinity of $\hookrightarrow$ also $M^{\beta}(u) + (\mu t^-)^{\beta}(u) \hookrightarrow M^{\beta}(u) + (\mu t^+)^{\beta}(u)$ holds. This yields $(M + \mu t^-)^{\beta}(u) \hookrightarrow (M + \mu t^+)^{\beta}(u)$ by linearity, which is what we wanted to show.

Now, by reflexivity and transitivity of $\hookrightarrow$ we get ${M_0}^{\beta}(u) \hookrightarrow M^{\beta}(u)$ for each reachable marking M of Σ. $\square$

We now express traditional notions as special cases of simulations.

Definition 17 (*Invariant expression*, *monotonic expression*). Let Σ be an algebraic system net, $\mathscr{M} = (B, +, 0, \hookrightarrow)$ a preordered commutative monoid such that $\mathscr{M}$ simulates Σ via u. Then, u is called

1. *$\mathscr{M}$-valued invariant expression* of Σ if $\hookrightarrow$ is an equivalence.
2. *$\mathscr{M}$-valued monotonic expression* of Σ if $\hookrightarrow$ is an order.

Definition 18 (*Place invariant*, *modulo-place invariant*). Let B be a set. An $\mathscr{M}$-valued invariant expression u of Σ is called

1. *place invariant* if $\mathscr{M} = \mathscr{L}_B(\mathbb{Z}, +, 0, =)$.
2. *modulo-k-place invariant* if $\mathscr{M} = \mathscr{L}_B(\mathbb{Z}, +, 0, \equiv_{\text{mod}\,k})$, where $\equiv_{\text{mod}\,k}$ denotes the residue class equivalence modulo k.

The expressiveness of invariant expressions, and therewith of place invariants, is quite restricted. Each invariant property that is implied by a linear invariant expression is preserved under reverse occurrence of transitions. Often however, central invariant properties of a system are not preserved under reverse occurrence of transitions. Therefore, invariant expressions are not sufficient for proving arbitrary invariant properties of the net. Classical supplementing techniques are traps and siphons which are special monotonic expressions. Here, we also introduce *semi-place invariants* which have not been considered so far for proving invariance properties. Subsequently, an example illustrates the benefit of using semi-place invariants.

Definition 19 (*Trap*, *siphon*, *semi-place invariant*). Let B be a set. An $\mathscr{M}$-valued monotonic expression u of Σ is called

1. (*individual*) *trap* if $\mathscr{M} = (2^B, \cup, \emptyset, \subseteq)$.
2. (*individual*) *siphon* if $\mathscr{M} = (2^B, \cup, \emptyset, \supseteq)$.
3. *increasing semi-place invariant* if $\mathscr{M} = \mathscr{L}_B(\mathbb{Z}, +, 0, \leqslant)$.

4. *decreasing semi-place invariant* if $\mathscr{M} = \mathscr{L}_{B'}(\mathbb{Z}, +, 0, \geqslant)$.

In Σ_1, we have, for example, the trap $supp(pr_1(\mathsf{messages}+\mathsf{distance}))$: Once there is a token with x as its first component at messages or distance, it remains so forever. Another trap of Σ_1 is $F(\mathsf{distance})$ where F is defined by $F(x,n) = \{(x,m) \mid m \geqslant n\}$. Treating $F(x,n)$ as a multiset, $F(\mathsf{distance})$ is even an increasing semi-place invariant.

From a trap, we can conclude that there is always a particular token at one of the corresponding places. An increasing semi-place invariant, however, has more potential: If it contains negative terms we may infer implications such as: If there is a particular token at place p then there is some other token at place q. Such a case is demonstrated in the following example.

Example 1 (*Semi-place invariant*). We consider Σ_1 again. We define functions F, G: $\mathbf{B}(A \times \mathbb{N}) \to \mathbb{Z}^{A \times \mathbb{N}}$ as linear extensions of $F, G\colon A \times \mathbb{N} \to \mathbb{Z}^{A \times \mathbb{N}}$ defined by:

$$F(x,n) = \{(x,m) \mid m \geqslant n\} \quad \text{and} \quad G(x,n) = M(x,n+1),$$

where M is the function that occurs in the inscription of Σ_1. Then, the linear expression $\mathsf{messages} + F(\mathsf{distance}) - G(\mathsf{distance})$ is an increasing semi-place invariant of Σ_1 and its initial value is []. First, we illustrate the verification of this statement. Subsequently, we derive an invariant proposition for Σ_1.

For verification we consider transition t3 as an example. Applying the substitutions $\mathsf{t3}^+$ and $\mathsf{t3}^-$ to the expression we get the following proof obligation:

$$n < m \Rightarrow (x,n) + F(x,m) - G(x,m) \leqslant M(x,n+1) + F(x,n) - G(x,n).$$

By definition of G this is equivalent to

$$n < m \Rightarrow (x,n) + F(x,m) \leqslant F(x,n) + G(x,m)$$

which holds true because, for $n < m$, we have

$$F(x,n) = [(x,n),(x,n+1),\ldots,(x,m-1)] + F(x,m).$$

The obligations for t1 and t2 can be verified similarily. We now show how Theorem 2 can be exploited.

By Theorem 2, we get that the following in-equation is satisfied:

$$[] \leqslant \mathsf{messages} + F(\mathsf{distance}) - G(\mathsf{distance})$$

which is equivalent to

$$G(\mathsf{distance}) \leqslant \mathsf{messages} + F(\mathsf{distance})$$

This in-equation on multisets is equivalent to the following propositions on elements:

$$\forall x,n: \quad G(\mathsf{distance})[(x,n)] \leqslant \mathsf{messages}[(x,n)] + F(\mathsf{distance})[(x,n)].$$

By definition of F we have $F(\mathsf{distance})[(x,n)] = \sum_{m \leqslant n} \mathsf{distance}[(x,m)]$ and by definition of G and M we have for all $(y,\ x) \in N$ the equation $G(\mathsf{distance})[(x,n)] = \mathsf{distance}[(y,n-1)]$. Together we get for all reachable markings:

$$\forall (y,x) \in N : \mathsf{distance}[(y,n-1)] \leqslant \mathsf{messages}[(x,n)] + \sum_{m \leqslant n} \mathsf{distance}[(x,m)]$$

This immediately implies the following invariance property: If agent y knows distance $n-1$ then each neighbour x has a message (x,n) or knows a distance $m \leqslant n$. This property is neither implied by any place invariant nor by any trap or siphon.

So far, we have considered linear expressions in order to prove invariance properties. Linear expressions can also be used for proving eventual properties such as ‘eventually some transition will not occur anymore’. Such eventual properties in turn help to prove more general eventual properties. We continue with introducing the notion of a stabilization expression.

Definition 20 (*Stabilization expression*, *termination expression*). Let $\mathscr{M} = (B, +, 0, \succcurlyeq)$ be a *regular* preordered commutative monoid, i.e. the monoid satisfies the following property:

$$\forall x, y, z \in B : x + z = y + z \Rightarrow x = y$$

Furthermore let u be a $\mathscr{M}$-valued monotonic expression of Σ.

1. A transition t of Σ is called *strict* with respect to u if

$$i(t) \Rightarrow t^-(u) \neq t^+(u)$$

2. u is called *stabilization expression* if $\succcurlyeq$ is well-founded. [7]
3. A stabilization expression is called *termination expression* if all transitions of Σ are strict with respect to it.

Theorem 3. *Let Σ be an algebraic system net, and let u be a $\mathscr{M}$-valued stabilization expression of Σ. Then, each process of Σ contains only finitely many occurrences of transitions that are strict with respect to u.*

Proof. Since $\mathscr{M}$ simulates Σ, for each $M_1 \xrightarrow{t,\mu} M_2$ holds $M_1(u) \succcurlyeq M_2(u)$ (see proof of Theorem 2). Moreover, we can show in the same manner that $M_1(u) \neq M_2(u)$ when t is strict w.r.t. u. This is because the contraposition of regularity is affinity of $\neq$. Since $\succcurlyeq$ is well-founded, we know that the value of u can be strictly decreased only finitely many times. Therefore, there can be only finitely many occurrences of strict transitions in a process. $\square$

[7] An order $\succcurlyeq$ is well-founded if there is no infinite strictly decreasing chain $x_0 \succ x_1 \succ x_2 \succ \dots$.

As a corollary we get: If there is a termination expression of Σ, then every process of Σ is finite. If we consider Σ_1 and choose the monoid $\mathbb{N} \times \mathbb{N}$ together with the lexicographic order, then $(|\mathsf{rootagents} + \mathsf{inneragents}|, SUM(pr_2(\mathsf{distance})))$ is a termination expression, where $SUM : \mathbf{B}(\mathbb{N}) \to \mathbb{N}$ denotes the sum of all elements of a bag.

Definition 21 (*Sur-place invariant*, *sub-place invariant*).

1. An increasing semi-place invariant is called *sur-place invariant* if all transitions are strict with respect to it.
2. A decreasing semi-place invariant is called *sub-place invariant* if all transitions are strict with respect to it.

If we have, in addition to a sur-place invariant (sub-place invariant), an upper (lower) bound for the expression, then we know that the system net always terminates. Proving termination in this way is sometimes more convenient than proving it by a termination expression as it allows negative terms.

6. Processes and unfoldings

In Section 2.5, we have defined the semantics of an algebraic system net in terms of its processes. In Section 4, we have defined the unfolding to a place/transition-system as an alternative semantics. On the other hand, there is a standard concept of processes for place/transition-systems [2]. Therefore, we have two different versions of processes of an algebraic system net: the processes of the direct definition and the processes of the unfolding.

In this last section, we demonstrate that both definitions coincide. To this end, we rephrase the definition of a process of a place/transition system, which mainly follows the line of [2].

Definition 22 (*Process of a place/transition system*). Let (P, T, W^-, W^+, M_0) be a place/transition system, $K = (B, E, F)$ be an occurrence net and $r : B \to P$ be a mapping. The pair (K, r) is a process of the place/transition-system if

1. for each place $p \in P$ holds $|\{b \in {}^\circ K \mid r(b) = p\}| = M_0[p]$ and
2. for each event $e \in E$ there exists a transition $t \in T$ such that, for each $p \in P$, we have $|\{b \in {}^\bullet e \mid r(b) = p)\}| = W^-(t)[p]$ and $|\{b \in e^\bullet \mid r(b) = p)\}| = W^+(t)[p]$.

Finally, we observe that each process of an algebraic system net is a process of its unfolding and vice versa.

Theorem 4. *Let Σ be an algebraic system net, and let K be an occurrence net. Then, (K, r) is a process of Σ if and only if (K, r) is a process of the unfolding $\widehat{\Sigma}$.*

Proof. Let $\Sigma = (N, \mathscr{A}, X, i)$ be an algebraic system net with net $N = (P, T, F)$, $K = (B, E, \lessdot)$ be an occurrence net and r be a condition labelling.

By definition, r is a mapping from B to $\widehat{P}$. Now, the two conditions for (K, r) being a process of Σ and $\widehat{\Sigma}$ can be shown to be equivalent, separately:

1. Suppose (K, r) satisfies condition 1 of Definition 11; i.e. $r(^\circ K) = M_0$. By definition of equality on multisets, this implies $r(^\circ K)(p)[a] = M_0(p)[a]$ for each ground sort $s \in GS$, each place $p \in P_{bs(s)}$, and each $a \in A_s$. By definition of $r(^\circ K)$, this implies $|\{b \in {^\circ K} \mid r(b) = p.a\}| = M_0(p)[a]$. By definition of $\widehat{P}$ and $\widehat{M_0}$, we get $|\{b \in {^\circ K} \mid r(b) = p.a\}| = \widehat{M_0}(p.a)$ for each $p.a \in \widehat{P}$. This is condition 1 of Definition 22.
   The reverse direction is similar.
2. Suppose (K, r) satisfies condition 2 of Definition 11; i.e. for each $e \in E$, there exists a transition $t \in T$ and a mode μ such that $\mu(i(t)) = \text{true}$, $r(^\bullet e) = \mu t^-$, and $r(e^\bullet) = \mu t^+$. By definition of $r(^\bullet e)$, the definition of μt^- and W^-, W^+ we have for each ground sort $s \in GS$, each place $p \in P_{bs(s)}$, and each $a \in A_s$: $r(^\bullet e)(p)[a] = |\{b \in {^\bullet e} \mid r(b) = p.a\}| = \mu t^-(p)[a] = \mu(i(p,t))[a] = W^-(t.\mu)[p.a]$. Similarly, we get $r(e^\bullet)(p)[a] = W^+(t.\mu)[p.a]$.
   This implies that for each $e \in E$ there exists a $\hat{t} \in \widehat{T}$ such that for each $\hat{p} \in \widehat{P}$ we have $|\{b \in {^\bullet e} \mid r(b) = \hat{p}\}| = W^-(\hat{t})[\hat{p}]$ and $|\{b \in e^\bullet \mid r(b) = \hat{p}\}| = W^+(\hat{t})[\hat{p}]$. This is condition 2 of Definition 22.
   The reverse direction is similar. □

7. Conclusion

In this paper, we have defined algebraic system nets along with a corresponding concept of place invariants. The main motivation was a net formalism for modelling distributed network algorithms. For the same reason, we have introduced a different syntactical representation of place invariants, viz. linear expressions, and their generalization to simulations. In particular, simulations turned out to be useful in the verification of distributed algorithms.

In this paper, we have concentrated on one particular kind of high-level nets, where markings of places are bags; i.e. elements of a free commutative monoid. Invariants for net types where markings are sets (P/E-systems) can be defined in the same way and give rise to similar verification conditions – we only have to change the allowed monoids for markings (cf. [17, 18, 5, 15]). Theorem 2 is valid for all net models where markings are elements of a commutative monoid.

Algebraic system nets are a generalization of algebraic Petri nets which overcomes some insufficiencies of the place invariant concept. Though inspired by the work of Vautherin [26] and Reisig [21], algebraic system nets as proposed in this paper show some fundamental differences:

1. There are no flexible arcs in [26, 21].
2. Reisig [21] uses algebraic specifications [8] for representing the involved algebra.

Here, we do not focus on that aspect; rather, we are free to use any appropriate formalism for representing the used algebra.

3. Reisig [21] represents a place invariant as a vector of terms. For convenience we represent a place invariant as a *linear expression* in which places may occur as variables. This representation was inspired by verification techniques for algebraic system nets, since linear expressions allow a smooth transition from Petri net concepts such as place invariants to temporal properties (cf. [22, 13, 27, 12]).
4. Reisig [21] introduces an occurrence rule as semantics for algebraic nets, only. In this paper we also introduce the non-sequential behaviour for algebraic system nets, which we call *processes* of the algebraic system net. This is justified, since we have shown that the set of processes of an algebraic system net exactly corresponds to the processes [2] of the unfolding.

Acknowledgements

We thank Karsten Schmidt, Wolfgang Reisig, and an anonymous referee for helpful suggestions and comments.

References

[1] T. Baar, E. Kindler, Hagen Völzer, Verifying intuition – ILF checks DAWN proofs, in: S. Donatelli, J. Kleijn (Eds.), Application and Theory of Petri Nets 1999, 20th Internat. Conf., Lecture Notes in Computer Science, vol. 1639, Springer, Berlin, June 1999, pp. 404–423.

[2] E. Best, C. Fernández, Nonsequential Processes, Springer, Berlin, 1988.

[3] J. Billington, Extending Coloured Petri Nets, Tech. Rep. 148, Computer Laboratory, University of Cambridge, Cambridge, October 1988.

[4] J. Billington, Extensions to coloured Petri nets and their application to protocols, Tech. Rep. No. 222, University of Cambridge, Cambridge, May 1991.

[5] B. Brosowski, A group-theoretical approach to Petri nets, Tech. Rep. 1998, preprint.

[6] M. Broy, On the design and verification of a simple distributed spanning tree algorithm, SFB-Bericht 342/24/90 A, Technische Universtität München, December 1990.

[7] J. Desel, K.-P. Neuendorf, M.-D. Radola, Proving nonreachability by modulo-invariants, Theoret. Comput. Sci. 153 (1996) 49–64.

[8] H. Ehrig, B. Mahr, Fundamentals of Algebraic Specifications 1, Equations and Initial Semantics, Springer, Berlin, 1985.

[9] U. Goltz, W. Reisig, The non-sequential behaviour of Petri nets, Inform. and Control 57 (1983) 125–147.

[10] K. Jensen, Coloured Petri Nets, vol. 2, Analysis Methods, Springer, Berlin, 1995.

[11] E. Kindler, W. Reisig, Algebraic system nets for modelling distributed algorithms, Petri Net Newslett. 51 (1996) 16–31.

[12] E. Kindler, W. Reisig, Verification of distributed algorithms with algebraic Petri nets, in: C. Freksa, M. Jantzen, R. Valk (Eds.), Foundations of Computer Science: Potential – Theory – Cognition, Lecture Notes in Computer Science, vol. 1337, Springer, Berlin, 1997, pp. 261–270.

[13] E. Kindler, W. Reisig, H. Völzer, R. Walter, Petri net based verification of distributed algorithms: an example, Formal Aspects Comput. 9 (1997) 409–424.

[14] E. Kindler, H. Völzer, Flexibility in algebraic nets, in: J. Desel, M. Silva (Eds.), Application and Theory of Petri Nets 1998, 19th Internat. Conf., Lecture Notes in Computer Science, vol. 1420, Springer, Berlin, June 1998, pp. 345–364.

[15] E. Kindler, M. Weber, The dimensions of Petri nets: the Petri net cube, EATCS Bull. 66 (1998) 155–166.
[16] G. Memmi, G. Roucairol, Linear algebra in net theory, in: W. Brauer (Ed.), Net Theory and Applications, Lecture Notes in Computer Science, vol. 84, Springer, Berlin, October 1979, pp. 213–223.
[17] J. Meseguer, U. Montanari, Petri nets are monoids, Inform. and Comput. 88 (1990) 105–155.
[18] J. Padberg, Abstract Petri nets: uniform approach and rule-based refinement, Ph.D. Thesis, Technische Universität Berlin, February 1996.
[19] J.L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[20] W. Reisig, Petri Nets, Springer, Berlin, 1985.
[21] W. Reisig, Petri nets and algebraic specifications, Theoret. Comput. Sci. 80 (1991) 1–34.
[22] W. Reisig, Petri net models of distributed algorithms, in: J. van Leeuwen (Ed.), Computer Science Today: Recent Trends and Developments, Lecture Notes in Computer Science, vol. 1000, Springer, Berlin, 1995, pp. 441–454.
[23] K. Schmidt, Verification of siphons and traps for algebraic Petri nets, in: P. Azéma, G. Balbo (Eds.), Application and Theory of Petri Nets 1997, 18th Internat. Conf., Lecture Notes in Computer Science, vol. 1248, Springer, Berlin, June 1997, pp. 427–446.
[24] E. Smith, W. Reisig, The semantics of a net is a net, an exercise in general net theory, in: K. Voss, H. Genrich, G. Rozenberg (Eds.), Concurrency and Nets. Springer, Berlin, 1987.
[25] R. Valk, Bridging the gap between place- and Floyd-invariants with applications to preemptive scheduling, in: M.A. Marsan (Ed.), Application and Theory of Petri Nets 1993, 14th Internat. Conf., Lecture Notes in Computer Science, vol. 691, Springer, Berlin, June 1993, pp. 431–452.
[26] J. Vautherin, Parallel systems specifications with coloured Petri nets and algebraic specifications, in: G. Rozenberg (Ed.), Advances in Petri Nets, Lecture Notes in Computer Science, vol. 266, Springer, Berlin, 1987, pp. 293–308.
[27] M. Weber, R. Walter, H. Völzer, T. Vesper, W. Reisig, S. Peuker, E. Kindler, J. Freiheit, J. Desel, DAWN: Petrinetzmodelle zur Verifikation Verteilter Algorithmen, Informatik-Bericht 88, Humboldt-Universität zu Berlin, December 1997.