Flexibility in Algebraic Nets

Ekkart Kindler and Hagen Völzer*

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

Abstract. *Algebraic Petri nets* as defined by Reisig [15] lack a feature for modelling distributed network algorithms, viz. *flexible arcs*. In this paper we equip algebraic Petri nets with flexible arcs and we 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 an unfolded algebraic Petri net which cannot be expressed as a place invariant of the algebraic Petri net itself. By introducing algebraic system nets along with a slightly more general concept of place invariants we also eliminate this insufficiency.
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, place invariants, verification techniques.

Introduction

Algebraic Petri nets as proposed by Reisig [15] lack a feature which is important 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 by help of an example. Then, we formally introduce a generalized version of algebraic Petri nets which allows for 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 the version in [15]. There, the unfolded algebraic Petri net may have a (low-level) place invariant which has no corresponding (high-level) place invariant in the algebraic Petri net. We will give an example for such a place invariant.

* supported by DFG: Konsensalgorithmen

For convenience, we do not use the traditional representation of a place invariant as a vector of weight functions [8] or a vector of terms [15]. Rather, we represent a place invariant as a *multiset-valued linear expression* in which place names may occur as bag-valued variables. Though this difference is only syntactical, it allows a smoother transition between Petri net properties and temporal logic (cf. [16, 11, 21, 10]). Moreover, it gives rise to a generalization: We can use expressions which 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 to more general domains has been proposed before (cf. [19, 5]); the use of affine preorders, however, is new. It turns out that well-known concepts like *siphons* (*deadlocks*) and *traps* [13, 14], *modulo-invariants* [5], and *sur-invariants* and *sub-invariants* ([12]) are special cases of *simulations*. Traps and siphons for algebraic Petri nets by Schmidt [17]. Modulo-invariants and sub- and sur-invariants for algebraic nets are introduced in this paper as the canonical adaption of the low-level versions. Moreover we introduce *semi-place-invariants* and *stabilization expressions* as further instances of simulations.

Also the use of flexible arcs in algebraic Petri nets is not completely new. Billington [2, 3] proposed some extensions which allow a restricted kind of 'flexibility' and Reisig [15] indicated some possible extensions. Our definition of algebraic system nets has been introduced in [9] — 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 [20] and Reisig [15] will be discussed in the conclusion.

The paper is organized as follows: In Sect. 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 Sect. 2 and their place invariants in Sect. 3. In Sect. 4 we define unfoldings of algebraic system nets and discuss the relation of place invariants of an unfolding with the place invariants of the algebraic system net itself. The generalization of place invariants to simulations will be defined in Sect. 5. Last, we show in Sect. 6 that processes of the unfolding are identical with 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 for each agent of the network the minimal distance from a root. This algorithm was inspired by a simple spanning tree algorithm [4]; the net model was already presented in [9] and verified in [10].

We denote the set of agents by A, the set of distinguished root-agents by $R \subseteq A$; the set of other so-called *inner agents* is denoted by $I = A \setminus R$. 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-

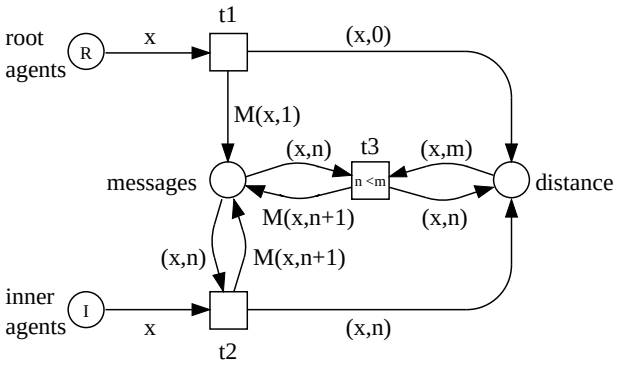

Fig. 1. A minimum distance algorithm Σ_1.

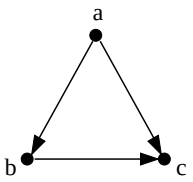

Fig. 2. A network of agents.

agent $x \in R$ sends a message to each of his *neighbours* in the network. In this message he informs his neighbours that they have distance 1 from a root (viz. from x himself). The agent $x \in R$ makes an entry for himself that his 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 he 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[1] $[(y_1, 1), \ldots, (y_n, 1)]$, where each pair represents a message to one neighbour.

An *inner agent* $x \in I$ waits until he receives a message from some of his neighbours. When he receives a message he accepts the distance n from this message; in addition he sends a message $n + 1$ to each of his neighbours. This behaviour is modelled by transition t2.

When an agent $x \in A$ receives another message with a distance n which is shorter than the distance m which he already knows, he accepts the new

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

distance n and sends the new distance $n+1$ to each of his neighbours. This behaviour is modelled by transition t3, where the transition guard guarantees $n < m$. Altogether, this behaviour guarantees that eventually each agent knows his 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 $n \in \mathbb{N}$. 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 t_1 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 [15].

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. Since sending messages to some neighbours is a primitive of network algorithms we should be able to represent it directly — without any tricks.

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, rootagents + inneragents + pr_1(distance). The function $pr_1 : A \times \mathbb{N} \to A$ is the projection of pairs to the first component, which is linearly extended to a function $pr_1 : \mathrm{BAG}(A \times \mathbb{N}) \to \mathrm{BAG}(A)$ in order to apply it to the bag distance.

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[2] $R + I = A$ 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 rootagents + inneragents + pr_1(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

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

the proposition rootagents+inneragents+pr_1(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.

For verifying that a linear expression is a place invariant of the system we have to check, for each transition, the validity of an equation. 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 rootagents + inneragents + pr_1(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 and right-hand side of the equation corresponding to a transition t will be denoted t^- and t^+ respectively. Then, a linear expression u is a place invariant of the algebraic system net, if for each transition t 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 [15]. Then, a place invariant is a vector i of multiset terms satisfying $N^T \cdot i = \underline{0}$, where the multiplication is term substitution. Our approach 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. [11]). 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 Sect. 4. Note, that this would also be possible in the vector notation.

1.3 More linear expressions

A place invariant is a linear expression of some multiset type. Its verification condition for each transition is $t^-(u) = t^+(u)$. Now let u be linear expression of any monoid type X, and let $\hookrightarrow \subseteq X \times X$ an affine[3] preorder in the monoid. Then we say u together with $\hookrightarrow$ *simulates* Σ if $t^-(u) \hookrightarrow t^+(u)$. If u evaluates to u_0 in the initial marking then we have $u_0 \hookrightarrow u$ for each reachable marking which allows the inference of invariance propositions.

For example, if we choose the monoid $(2^A, \cup, \emptyset)$ and the preorder $\supseteq$ then *supp* rootagents $\cup$ *supp* inneragents is a linear expression which simulates Σ_1, 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* rootagents $\cup$ *supp* 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*

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

will be formalized similarily. Moreover, we introduce *semi-place-invariants* and *stabilization expressions* as further useful instances of simulations.

A stabilization expression simulates an algebraic system net together with a well-founded affine order. Transitions which strictly decrease the value of the expression can happen only finitely many times: In each run eventually a marking is reached such that all those transitions are disabled forever. A special case of stabilization is termination: A *termination expression* proves that in each run a deadlock is reached. Sometimes, in Petri net theory, *sur-place-invariants* and *sub-place-invariants* [12] 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 $t^-(u) \hookrightarrow t^+(u)$. This is the main benefit of this approach.

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 algebraic specifications [6] and Petri nets [14]. The only new concept is the *bag-signature* together with a corresponding concept of a *bag-algebra*.

Sets, families, and functions. By $\mathbb{B}, \mathbb{N}, \mathbb{Z}$ we denote the set $\{\text{true}, \text{false}\}$ 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 $|A|$, denote the set of all non-empty finite sequences over A by A^+, and 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*, iff for each $i, j \in I$ with $i \neq j$ holds $A_i \cap A_j = \emptyset$. If $A = (A_i)_{i\in I}$ is a family of sets, then the set $\bigcup_{i\in I} A_i$ is often also denoted by A, for convenience. For two sets A and B we denote the set of all mappings from A to B by $B^A = \{f \mid f : A \to B\}$. If we have $f_1 : A \to B$ and $f_2 : C \to D$ such that A and C are disjoint then $(f_1 \uplus f_2) : A \cup C \to B \cup D$ denotes the union of both functions.

Monoids. A set A together with a commutative associative binary operation $+$ and a neutral element 0 is called *commutative monoid*; if there is additionally a reflexive and transitive relation $\hookrightarrow \subseteq A \times A$, then we call $\mathcal{M} = (A, +, 0, \hookrightarrow)$ *preordered commutative monoid* iff $\hookrightarrow$ is *affine*, i.e. iff $\forall x, y, z \in A : x \hookrightarrow y \implies x + z \hookrightarrow y + z$.

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

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 $\leq$ of multisets by lifting $(\mathbb{Z}, +, 0, \leq)$ over A. The *support* of a multiset is defined by $supp\, M = \{x \in A \mid M[x] \neq 0\}$. M is *nonnegative* iff $M[x] \geq 0$ for all x in A, and M is *finite* iff $supp\, M$ is finite.

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

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* $\mathcal{A} = ((A_s)_{s \in S}, (f_{op})_{op \in OP})$ 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 $op \in OP_{s_1 \ldots s_n\, s_{n+1}}$ we have $f_{op} : A_{s_1} \times \ldots \times A_{s_n} \to A_{s_{n+1}}$. A set A_s of an algebra is called *domain* and a function f_{op} is called *operation* of the algebra.

In the following we assume that a signature SIG has a sort symbol $bool \in S$ and in each SIG-algebra the corresponding domain is $A_{bool} = \mathbb{B}$.

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*. A *term* is built up from variables and operation symbols. 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. $u_i \in \mathbf{T}_{s_i}^{SIG}(X)$ for $i = 1, \ldots, n$ and $op \in OP_{s_1 \ldots s_n\, s_{n+1}}$ implies $op(u_1, \ldots, u_n) \in \mathbf{T}_{s_{n+1}}^{SIG}(X)$

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 $\mathbf{T}_s^{SIG} = \mathbf{T}_s^{SIG}(\emptyset)$.

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

1. $\overline{\beta}(x) = \beta(x)$ for $x \in X$.
2. $\overline{\beta}(op(u_1, \ldots, u_n)) = f_{op}(\overline{\beta}(u_1), \ldots, \overline{\beta}(u_n))$ for $op(u_1, \ldots, u_n) \in \mathbf{T}^{SIG}(X)$.

The mapping $\overline{\beta}$ is called *β-evaluation in $\mathcal{A}$*. Let $\underline{\beta_\emptyset} : \emptyset \to A$ be the unique assignment for the empty variable set. By $\mathrm{eval} := \overline{\beta_\emptyset}$ we denote the evaluation of ground terms.

Substitutions. Let X and Y be SIG-variable sets. A mapping $\sigma : X \to \mathbf{T}^{SIG}(Y)$ is called *substitution* iff $x \in X_s$ implies $\sigma(x) \in \mathbf{T}_s^{SIG}(Y)$. Analogously to evaluations, we also extend σ to a mapping $\overline{\sigma} : \mathbf{T}^{SIG}(X) \to \mathbf{T}^{SIG}(Y)$ in order to apply it to terms. In case of $Y = \emptyset$ we call σ *ground substitution.*

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 1 (Bag-signature, $BSIG$-algebra). Let $SIG = (S, OP)$ be a signature and $GS, BS \subseteq S$; $BSIG = (S, OP, bs)$ is a *bag-signature* iff $bs : GS \to BS$ 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 $\mathcal{A}$ is a *$BSIG$-algebra* iff for each $s \in GS$ holds $A_{bs(s)} = \mathrm{BAG}(A_s)$, i.e., if for each *ground-domain* the corresponding *bag-domain* is actually the set of all bags over the ground-domain.

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, terms, assignments, evaluation, and substitutions are well-defined for bag-signatures, too.

2.2 Algebraic system nets

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 iff both, P and T, are finite.

Definition 2 (Place/transition system).
A *place/transition system* $\Sigma = (N, W, M_0)$ consists of

1. a net $N = (P, T, F)$,
2. a *weight function* $W : F \to \mathbb{N}$, and
3. a *marking* M_0, called *initial marking* of Σ, where a marking of a place/transition system is a mapping $M : P \to \mathbb{N}$.

We extended W to $W : (P \cup T) \times (T \cup P) \to \mathbb{N}$ by $W(f) = 0$ if $f \notin F$.

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

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 $p \in P_s : i(p) \in \mathbf{T}_s^{BSIG}$, i.e., the restriction of i to P is a ground substitution for P,
   (b) for each $t \in T : i(t) \in \mathbf{T}_{bool}^{BSIG}(X)$, 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 $i(f) \in \mathbf{T}_s^{BSIG}(X)$.

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.

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

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

In a sense, Def. 3 gives the syntax of an algebraic system net. The algebra is still given semantically because we want to be flexible. We can incorporate any appropriate formalism for representing an algebra. The semantics, i.e. the processes of an algebraic system net, will be defined in Sect. 2.3. Here, we define *markings* and the *firing-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 Def. 3. A *marking M of Σ* is an assignment for P. The marking M_0 with $M_0(p) = \text{eval}(i(p))$ 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 fire 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 fire in mode β, 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 firing-rule by associating each pair (t,β) with a marking t_β^- and a marking t_β^+. The marking t_β^- and the marking t_β^+ represent the elements which are removed and added respectively, when t fires in mode β.

Definition 6 (Firing rule and reachable markings). Let Σ be an algebraic system net as in Def. 3. Let $t \in T$ and β be an assignment of X in $\mathcal{A}$. We define the two markings t_β^- and t_β^+ by $t_\beta^-(p) = \overline{\beta}(t^-(p))$ and $t_\beta^+(p) = \overline{\beta}(t^+(p))$.

In a given marking M_1 a transition t is *enabled* in mode β, iff there exists a marking M such that $M_1 = M + t_\beta^-$ and $\overline{\beta}(i(t)) = \text{true}$. Then, transition t may fire in mode β, which results in the *successor marking* $M_2 = M + t_\beta^+$. We

denote the firing of transition t in mode β by $M_1 \stackrel{t,\beta}{\rightarrow} M_2$. We say a marking M' is *reachable from* a marking M, denoted $M \stackrel{*}{\rightarrow} M'$, iff there exists a finite chain of markings $M_1, \ldots, M_n$ such that $M_1 = M$, $M_n = M'$, and for each i the marking M_{i+1} is a successor marking of M_i. We say that a marking M is a *reachable marking of* Σ iff M is reachable from M_0, i.e. the initial marking of Σ.

Remark 7. In the following we only consider algebraic system nets in which for each transition t and each mode β, the markings t_β^- and t_β^+ are nonempty. This helps to avoid some anomalies in the definition of processes (see [1] for further explanation).

2.3 Processes of algebraic system nets

Now we define non-sequential processes [7, 1] for algebraic system nets. Figure 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 firing of a transition of the algebraic system net in some mode.

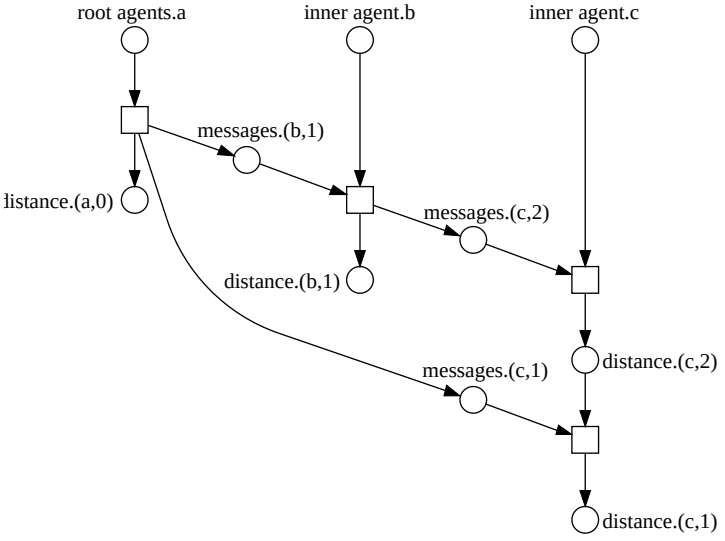

Fig. 3. A process of Σ_1.

For the formal definition of processes we start with some notations and definitions which mainly follow the lines of [1].

Definition 8. 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 $\downarrow x = \{y \in P \cup T \mid (y,x) \in F^+\}$, where F^+ denotes the transitive closure[4] of the flow relation F.

Processes are defined by 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 [7, 1].

Definition 9 (Occurrence net). A net $K = (B, E, <)$ 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 $|{}^\bullet b| \leq 1$ and $|b^\bullet| \leq 1$, 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 new symbols for places and transitions of an occurrence net. Moreover, we call a place of an occurrence net *condition* and a transition *event*. Next we define the *states* of an occurrence net.

Definition 10 (States of an occurrence net). Let $K = (B, E, <)$ be an occurrence net. For subsets of conditions $Q, Q' \subseteq B$ we define the occurrence relation $\rightarrow$ by: $Q \rightarrow Q'$ iff there exists an event $e \in E$ such that ${}^\bullet e \subseteq Q$ and $Q' = (Q \setminus {}^\bullet e) \cup e^\bullet$. The transitive and reflexive closure of $\rightarrow$ is denoted by $\stackrel{*}{\rightarrow}$. For $Q, Q' \subseteq B$ we say Q' *is reachable from* Q, if $Q \stackrel{*}{\rightarrow} Q'$.

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.

Definition 11 (Condition labelling). Let Σ be an algebraic system net over a bag-signature *BSIG* as in Def. 3, and $K = (B, E, <)$ be an occurrence net. A mapping $r : B \rightarrow P \times A$ is a *condition labelling of* K, iff 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 $r(Q) : P \rightarrow \mathrm{BAG}(A)$ 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 firing of a transition in some mode (cf. Fig. 3).

[4] Note, that we do not use the transitive and reflexive closure of F. This way, we can express acyclicity of F by $p \notin \downarrow p$ for each place $p \in P$.

Definition 12 (Process). Let Σ be an algebraic system net, $K = (B, E, <)$ be an occurrence net, and r be a condition labelling of K. Then, (K, r) is a *process* of Σ, iff

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 β such that $\overline{\beta}(i(t)) = \text{true}$, $r(^\bullet e) = t^-_\beta$, and $r(e^\bullet) = t^+_\beta$.

Def. 12 is the canonical extension of processes [1] to algebraic system nets (see also Sect. 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 linear expressions rather than vectors of terms for representing place invariants. In these expressions places are interpreted as variables of the corresponding bag sort.

Definition 13 (Place invariant). Let *BSIG* be a bag-signature, Σ be an algebraic system net over *BSIG* with places P. Let B be some set and $v \in \mathbf{T}^{BSIG}_{\mathbb{Z}^B}(P)$ a multiset-valued expression with the place names of the net as variables. Given a marking of Σ, i.e. an assignment for P, expression v can be evaluated to $v_M := \overline{M}(v)$; then, v is called *place invariant* of Σ iff

1. v is linear, i.e. for all M_1, M_2 it holds that $v_{M_1+M_2} = v_{M_1} + v_{M_2}$, and
2. for all transitions t the conditional equation $i(t) \implies t^-(v) = t^+(v)$ holds.

Remark 14. Since $\mathbb{Z}^{\{\bullet\}}$ is isomorphic to $\mathbb{Z}$ we also take integer valued expression into consideration for place invariants. We call such a place invariant *simple*.

Note that we defined linearity semantically. A syntactical characterization is straight-forward and can be found in [21]. As already stated, the evaluation of a place invariant is constant for all reachable markings:

Theorem 15. *If v is a place invariant of Σ and M_0 its initial marking then all reachable markings M of Σ satisfy $v_M = v_{M_0}$.*

Proof. Consequence of the forthcoming Theorem 21.

Reisig [15] represents a place invariant by a P-vector of multiset terms: To each place $p \in P$ a non-flexible multiset term is assigned, which represents a function f_p. Here non-flexibility means: For all markings M_1, M_2 we have $|M_1(p)| = |M_2(p)|$ implies $|f_p(M_1(p))| = |f_p(M_2(p))|$. 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 [15] translates to the linear expression $f_{p_1}(p_1) + f_{p_2}(p_2) + \cdots + f_{p_n}(p_n)$.

In Section 5 we redefine place invariants where we allow also additional variables in the expression (Such variables are also allowed in the terms f_p of [15]). Such variables do not increase the expressivity of place invariants, but the use of them is sometimes convenient.

4 Unfoldings

In Sect. 2.3 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. [18]). 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.

First, we will present the definition of an unfolding. Subsequently, we give an example of an algebraic Petri net [15] which has a place invariant in the unfolding but no corresponding place invariant (according to the definition of [15]) 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. 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 β; 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 16 (Unfolding). Let $\Sigma = (N, \mathcal{A}, X, i)$ 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. $\widehat{P} = \bigcup_{s \in GS} \bigcup_{p \in P_{bs(s)}} \{(p, a) \mid a \in A_s\}$
2. $\widehat{T} = \bigcup_{t \in T} \{(t, \beta) \mid \overline{\beta}(i(t)) = \text{true}\}$
3. $\widehat{F} \subseteq (\widehat{P} \times \widehat{T}) \cup \widehat{T} \times \widehat{P}$ by
   $\widehat{F} = \{((p, a), (t, \beta)) \mid (p, a) \in \widehat{P}, (t, \beta) \in \widehat{T}, \overline{\beta}(i(p, t))[a] \geq 1\} \cup$
   $\{((t, \beta), (p, a)) \mid (p, a) \in \widehat{P}, (t, \beta) \in \widehat{T}, \overline{\beta}(i(t, p))[a] \geq 1\}$
4. $\widehat{W} : \widehat{F} \to \mathbb{N}$ by $\widehat{W}((p, a), (t, \beta)) = \overline{\beta}(i(p, t))[a]$ resp. $\widehat{W}((t, \beta), (p, a)) = \overline{\beta}(i(t, p))[a]$ for $(p, a) \in \widehat{P}$, $(t, \beta) \in \widehat{T}$.
5. $\widehat{M_0} : \widehat{P} \to \mathbb{N}$ by $\widehat{M_0}((p, a)) = M_0(p)[a]$ for $(p, a) \in \widehat{P}$.

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

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

4.2 Place Invariants of Unfoldings

In this section we will show that each place invariant of the unfolding can be represented as a place invariant of the original algebraic system net. Though, this is a desirable property and seems to be obvious, this does not hold for place invariants of Reisig [15]; this will be demonstrated by a simple example. Vice versa, each place invariant of the algebraic system net represents a set of place invariants in the low-level system. First, we present the traditional definition of place invariants of a place/transition-system.

Definition 17 (Place invariants of place/transition-systems).
Let (N, W, M_0) be a place/transition-system with $N = (P, T, F)$. A mapping (often called vector in this context) $j : P \to \mathbb{Z}$ is a *place invariant* of the place/transition-system, if for each transition $t \in T$ the following equation holds:

$$\sum_{p \in P} j(p) \cdot W(p, t) = \sum_{p \in P} j(p) \cdot W(t, p).$$

The main idea of a place invariant j is that it can be interpreted as a weighting of markings by $j(M) = \sum_{p \in P} j(p) \cdot M(p)$. Then, we have for each two markings with $M \longrightarrow M'$: $j(M) = j(M')$.

A counter-example. First of all, we demonstrate that in the formalism of Reisig [15] there exists a place invariant of an unfolded algebraic Petri net which has no corresponding place invariant in the algebraic Petri net itself. Consider the algebraic Petri net Σ_2 of Fig. 4, where a and b are two different constants of the same sort. Figure 5 shows the unfolding $\widehat{\Sigma_2}$ of this algebraic system net. Obviously, $j = (p_1, a) + (p_2, a)$ is a place invariant of $\widehat{\Sigma_2}$. Now, we will show that Σ_2 has no place invariant which corresponds to j, when we restrict 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 $f_1(p_1)+f_2(p_2)$. It follows that $|f_1(p_1) + f_2(p_2)|$ is also a place invariant of Σ_2 which can equivalently be rewritten to $|f_1(p_1)| + |f_2(p_2)|$. Since the invariant u was of non-flexible, we know that there exist integer values n_1 and n_2 such that $|f_1(p_1)|+|f_2(p_2)| = n_1 \cdot |p_1| + n_2 \cdot |p_2|$. 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 place invariant which evaluates to 0 for each marking; i.e. u is a trivial place invariant.

The reason why there are only trivial place invariants of Σ_2 in the approach of Reisig [15] is that each token on a place is mapped to a multiset of the same

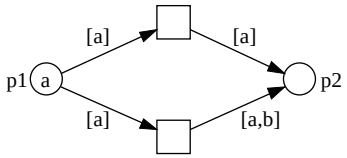

Fig. 4. An algebraic system net Σ_2.

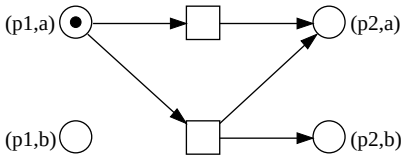

Fig. 5. The unfolding $\widehat{\Sigma_2}$.

cardinality. In order to express the invariant j of the unfolding, it is necessary to map a token a on places p_1 and p_2 to a singleton multiset (e.g. by $[a]$) 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 $f_b([b]) = []$, where f_a is not a legal function in the approach of [15].

Correspondence of place invariants. Next, we will see that allowing flexibility in arcs as well as in place invariants gives an exact correspondence of place invariants of an algebraic system net with the place invariants of its unfolding. Of course, this correspondence depends on the expressiveness of the underlying algebra. Here, we allow to extend an algebra by appropriate operators. In order to formalize this result, we must formalize when a place invariant of an algebraic system net corresponds to a place invariant of the unfolding.

Since a place invariant j of the unfolding $\widehat{\Sigma}$ evaluates to $\mathbb{Z}$, we actually have a simple place invariant of the algebraic system net Σ corresponding to j (cf. Remark 14). Therefore, we define *correspondence of place invariants* for simple place invariants, only.

Definition 18 (Correspondence of place invariants). Let Σ be an algebraic system net with places P and $u \in \mathbf{T}_{\mathbb{Z}}^{BSIG}(P)$ be a simple place invariant of Σ, and j be a place invariant of $\widehat{\Sigma}$. We say u corresponds to j, if for each marking M of Σ holds $u_M = \sum_{\hat{p} \in \hat{P}} \widehat{M}(\hat{p}) \cdot j(\hat{p})$.

Now, we fix an arbitrary place invariant j of the unfolding $\widehat{\Sigma}$. We define for each place $p \in P_s$ of Σ the operation $f_p^j : A_s \to \mathbb{Z}$ as linear extension of $f_p^j([a]) = j(p, a)$ for each $a \in A_s$. Then, $f_{p_1}^j(p_1) + \ldots + f_{p_n}^j(p_n)$ is a simple place invariant of Σ corresponding to j, where we assume that $P = \{p_1, \ldots, p_n\}$ are the places of Σ.

Theorem 19. *Let Σ be an algebraic system net with finitely many places and j be a place invariant of the unfolding $\widehat{\Sigma}$. Then there exists a simple place invariant u of Σ which corresponds to j.*

Proof. We choose $u = f_{p_1}^j(p_1) + \ldots + f_{p_n}^j(p_n)$. Obviously, u and j correspond to each other. It remains to be shown that u is a place invariant of Σ. We prove the contraposition. Let us assume that u is not a place invariant of Σ; now, we show that j is not a place invariant of $\widehat{\Sigma}$.

Since u is not a place invariant of Σ there exists a transition such that the implication $i(t) \implies t^-(u) = t^+(p)$ is not valid. Therefore, there exists an assignment β such that $\overline{\beta}(i(t)) = \text{true}$ and $\overline{\beta}(t^-(u)) \neq \overline{\beta}(t^+(p))$. By definition of $\widehat{\Sigma}$ there exists a transition $\hat{t} = (t, \beta) \in \widehat{T}$ in the unfolding $\widehat{\Sigma}$. Then, we have

$\overline{\beta}(t^-(u)) =$	def. of u and t^-
$\overline{\beta}(f^j_{p_1}(i(t,p_1)) + \ldots + f^j_{p_n}(i(t,p_n))) =$	def. of $\overline{\beta}$
$f^j_{p_1}(\overline{\beta}(i(t,p_1))) + \ldots + f^j_{p_n}(\overline{\beta}(i(t,p_n))) =$	equiv. multiset-represent.
$f^j_{p_1}(\sum_{a\in A} \overline{\beta}(i(t,p_1))[a] \cdot [a]) + \ldots +$	
$\quad f^j_{p_n}(\sum_{a\in A} \overline{\beta}(i(t,p_n))[a] \cdot [a]) =$	linearity of functions f^j_p
$\sum_{a\in A} \overline{\beta}(i(t,p_1))[a] \cdot f^j_{p_1}([a]) + \ldots +$	
$\quad \sum_{a\in A} \overline{\beta}(i(t,p_n))[a] \cdot f^j_{p_n}([a]) =$	
$\sum_{a\in A} \sum_{p\in P} \overline{\beta}(i(t,p))[a] \cdot f^j_p([a]) =$	def. of W and f^j_p
$\sum_{a\in A} \sum_{p\in P} W((t,\beta),(p,a)) \cdot j(p,a) =$	def. of $\widehat{P}$
$\sum_{\hat{p}\in\widehat{P}} W(\hat{t},\hat{p}) \cdot j(\hat{p})$	

By the same arguments we get $\overline{\beta}(t^+(u)) = \sum_{\hat{p}\in\widehat{P}} W(\hat{p},\hat{t}) \cdot j(\hat{p})$. Then by $\overline{\beta}(t^-(u) \neq \overline{\beta}(t^+(p))$ we have $\sum_{\hat{p}\in\widehat{P}} W(\hat{t},\hat{p}) \cdot j(\hat{p}) \neq \sum_{\hat{p}\in\widehat{P}} W(\hat{p},\hat{t}) \cdot j(\hat{p})$, which implies that j is not a place invariant of $\widehat{\Sigma}$.

Now, we consider the reverse direction. We proceed in two steps. First, we show that each simple place invariant of an algebraic system net corresponds to a place invariant of the unfolding. Then, we show that each place invariant of an algebraic system net corresponds to a family of simple place invariants of the algebraic system net.
Let u be a simple place invariant of Σ. We define $j : \widehat{P} \to \mathbb{Z}$ such that $j(p,a) = \overline{\beta}_{(p,a)}(u)$ where $\beta_{(p,a)}$ is the assignment defined by $\beta_{(p,a)}(p) = [a]$ and $\beta_{(p,a)}(q) = []$ for $q \neq p$. Then, j is a place invariant of $\widehat{\Sigma}$ which corresponds to u.
Now, let u be a place invariant with domain $\mathbb{Z}^B$ of the algebraic system net Σ. Then, for each $b \in B$ the expression $u[b]$ is a simple place invariant of Σ. Therefore, place invariant u can equivalently be represented by the family $(u[b])_{b\in B}$ of simple place invariants.
Altogether, we get that a place invariant of Σ corresponds to a family of place invariants of the unfolding $\widehat{\Sigma}$.

5 More linear verification techniques

Now we define linear expressions which is the basic notion of this section. Subsequently we define several known and some new linear verification techniques as special linear expressions.

Definition 20 (Expressions, linear expressions). Let *BSIG* be a bag-signature with sorts S and let Σ be an algebraic system net over *BSIG* with places P and variables X. Furthermore let Y be a variable set disjoint from P and X. A *Σ-expression* $(u : \mathcal{M})$ consists of a term $u \in \mathbf{T}_s^{BSIG}(Y \cup P)$ and a preordered commutative monoid $\mathcal{M} = (A_s, +, 0, \hookrightarrow)$, called the *type* of the Σ-expression.

Given an assignment γ for Y and an assignment M for P (i.e. a marking) the term u evaluates to $u_M^\gamma := \overline{\gamma \uplus M}(u)$. A Σ-expression $(u : \mathcal{M})$ is *linear* iff:

$$\forall \gamma : \forall M_1, M_2 : u^\gamma_{M_1+M_2} = u^\gamma_{M_1} + u^\gamma_{M_2} \tag{1}$$

We say $(u : \mathcal{M})$ *simulates* Σ (is a *simulation of* Σ) iff for each transition t of Σ the following condition is satisfied:

$$i(t) \implies t^-(u) \hookrightarrow t^+(u) \tag{2}$$

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 21. *Let Σ be an algebraic system net with initial state M_0, $(u : \mathcal{M})$ a linear Σ-expression which simulates Σ. Then, for each assignment γ and for each reachable marking M of Σ we have $u^\gamma_{M_0} \hookrightarrow u^\gamma_M$.*

Proof. Let γ be an arbitrary assignment. First we show that $M \overset{t,\beta}{\to} M'$ implies $u^\gamma_M + u^\gamma_{M'}$ for all markings M, M' of Σ: If we have $M \overset{t,\beta}{\to} M'$ then we have $\overline{\beta}(i(t)) = \text{true}$ and it exists a marking $\widetilde{M}$ such that $M = \widetilde{M} + t^-_\beta$ and $M' = \widetilde{M} + t^+_\beta$. By (2) we get $\overline{\beta \uplus \gamma}(t^-(u)) \hookrightarrow \overline{\beta \uplus \gamma}(t^+(u))$ and therefore $u^\gamma_{t^-_\beta} \hookrightarrow u^\gamma_{t^+_\beta}$. By affinity of $\hookrightarrow$ also $u^\gamma_{\widetilde{M}} + u^\gamma_{t^-_\beta} \hookrightarrow u^\gamma_{\widetilde{M}} + u^\gamma_{t^+_\beta}$ holds. This yields $u^\gamma_{\widetilde{M}+t^-_\beta} \hookrightarrow u^\gamma_{\widetilde{M}+t^+_\beta}$ by linearity (1) which is what we wanted to show.

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

We now derive traditional notions as special cases of simulations.

Definition 22 (Invariant expression, monotonic expression). Let Σ be an algebraic system net, $\mathcal{M} = (B, +, 0, \hookrightarrow)$ a preordered commutative monoid and $(u : \mathcal{M})$ a linear Σ-expression which simulates Σ. Then, $(u : \mathcal{M})$ is called

1. *invariant expression* of Σ iff $\hookrightarrow$ is an equivalence.
2. *monotonic expression* of Σ iff $\hookrightarrow$ is an order.

Definition 23 (Place invariant, modulo-place-invariant). Let B be a set. An invariant expression $(u : \mathcal{M})$ of Σ is called

1. *place invariant* iff $\mathcal{M} = \mathcal{L}_B(\mathbb{Z}, +, 0, =)$.

2. *modulo-k-place-invariant* iff $\mathcal{M} = \mathcal{L}_B(\mathbb{Z}, +, 0, \equiv_{\bmod k})$, where $\equiv_{\bmod k}$ denotes the remainder class equivalence.

The expressiveness of invariant expressions is quite restricted. They only imply invariant properties which are preserved under reverse firing. If a desired invariant property is not derivable from an invariant expression, a monotonic expression might help.

Definition 24 (Trap, siphon, semi-place-invariant).
Let B be a set. A monotonic expression $(u : \mathcal{M})$ of Σ is called

1. *(individual) trap* iff $\mathcal{M} = (2^B, \cup, \emptyset, \subseteq)$.
2. *(individual) siphon* iff $\mathcal{M} = (2^B, \cup, \emptyset, \supseteq)$.
3. *increasing semi-place-invariant* iff $\mathcal{M} = \mathcal{L}_B(\mathbb{Z}, +, 0, \leq)$.
4. *decreasing semi-place-invariant* iff $\mathcal{M} = \mathcal{L}_{B'}(\mathbb{Z}, +, 0, \geq)$.

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 $F(x,n) = \{(x,m) \mid m \geq n\}$. Treating $F(x,n)$ as a multiset, $F(\mathsf{distance})$ is even an increasing semi-place-invariant.

From a trap we may only conclude that there is 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 directly 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 25 (Semi-place-invariant). Let $\chi B' : B \to \{0,1\}$ denote the characteristic function of a set $B' \subseteq B$ which is defined by $(\chi B')(b) = 1$ iff $b \in B'$. We consider $\chi B'$ also as a multiset over B. We consider Σ_1 again. We define functions $F, G : \mathrm{BAG}(A \times \mathbb{N}) \to \mathbb{Z}^{A \times \mathbb{N}}$ as linear extensions of $F, G : A \times \mathbb{N} \to \mathbb{Z}^{A \times \mathbb{N}}$ defined by:

$$F(x,n) = \chi\{(x,m) \mid m \geq n\} \qquad \text{and} \qquad G(x,n) = M(x,n+1)$$

where M is the function which 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 $[]$. 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:

$$n < m \implies (x,n) + F(x,m) - G(x,m) \leq M(x,n+1) + F(x,n) - G(x,n)$$

By definition of G this is equivalent to

$$n < m \implies (x,n) + F(x,m) \leq F(x,n) + G(x,m)$$

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

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

Now we can conclude by Theorem 21 that the following in-equation is satisfied:

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

which is equivalent to

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

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

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

By definition of F we have $F(\mathsf{distance})[(x,n)] = \sum_{m \leq n} \mathsf{distance}[(x,m)]$ 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:

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

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 $m \leq n$.

Next we investigate some verification techniques for special liveness properties of an algebraic system net.

Definition 26 (Stabilization expression, termination expression).
Let $\mathcal{M} = (B,+,0,\preceq)$ be a *regular* preordered commutative monoid, i.e. the monoid satisfies

$$\forall x,y,z \in B : x+z = y+z \implies x = y \tag{3}$$

Furthermore let $(u : \mathcal{M})$ be a monotonic expression of Σ.

1. A transition t of Σ is called *strict* with respect to $(u : \mathcal{M})$ iff

$$i(t) \implies t^-(u) \neq t^+(u) \tag{4}$$

2. $(u : \mathcal{M})$ is called *stabilization expression* iff $\preceq$ is well-founded[5].

[5] An order $\prec$ is well-founded iff there is no infinite strictly decreasing chain $x_0 \prec x_1 \prec x_2 \prec \ldots$.

3. A stabilization expression is called *termination expression* iff all transitions of Σ are strict with respect to it.

Theorem 27. *Let Σ be an algebraic system net and $(u : \mathcal{M})$ a stabilization expression of Σ. Then, each process of Σ contains only finitely many occurrences of transitions which are strict w.r.t. $(u : \mathcal{M})$.*

Proof. Since $(u : \mathcal{M})$ is a simulation for each $M \stackrel{t,\beta}{\rightarrow} M'$ holds $u_M \preceq u_{M'}$ (see proof of Theorem 21). Moreover we can show in the same manner that $u_M \neq u_{M'}$ when t is strict w.r.t. $(u : \mathcal{M})$. This is because the contraposition of (3) is affinity of $\neq$. Since $\preceq$ 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.

As a corrolary 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 $SUM : \mathrm{BAG}(\mathbb{N}) \rightarrow \mathbb{N}$ denotes the sum of all elements of a bag.

Definition 28 (Sur-place-invariant, sub-place-invariant).

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

If we have in addition to a sur-place-invariant (sub-place-invariant) also a higher (lower) bound for the expression then we know that the system always terminates. Proving termination this way is sometimes more convenient than proving it by a termination expression as it allows negative terms, i.e. the use of the difference, in the expression. Sur/sub-place invariants were introduced in [12].

6 Processes and Unfoldings

In Sect. 2.3 we have defined the semantics of an algebraic system net in terms of its processes. In Sect. 4 we have defined the unfolding to a place/transition-system as an alternative semantics. Now, there is a standard concept of processes for place/transition-systems [1]. 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 will 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 [1].

Definition 29 (Process of a place/transition system).
Let $((P,T,F),W,m)$ 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, iff

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

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

Theorem 30. *Let Σ be an algebraic system net, K be an occurrence net. Then, (K,r) is a process of Σ, if and only if (K,r) it is a process of the unfolding $\widehat{\Sigma}$.*

Proof. Let $\Sigma=(N,\mathcal{A},X,i)$ be an algebraic system net with net $N=(S,T,F)$, $K=(B,E,\lessdot)$ 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 Def. 12; i.e. $r({}^\circ K)=M_0$. By definition of equality on multisets this is implies $r({}^\circ K)(p)[a]=M_0(p)[a]$ for each sort $s \in S$, each place $p \in P_s$, and each $a \in A_{bs^{-1}(p)}$. 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 Def. 29.
   The reverse direction ist similar.
2. Suppose (K,r) satisfies condition 2 of Def. 12; i.e. for each $e \in E$ there exists a $t \in T$ and an assignment β such that $\overline{\beta}(t)=true$, $r({}^\bullet e)=t_\beta^-$, and $r(e^\bullet)=t_\beta^+$.
   By definition of $r({}^\bullet e)$, the definition of t_β^-, and the definition of $\widehat{W}$ we have for each sort $s \in S$, each place $p \in P_s$, and each $a \in A_{bs^{-1}(p)}$: $r({}^\bullet e)(p)[a] = |\{b \in {}^\bullet e \mid r(b)=(p,a)\}| = t_\beta^-(p)[a] = \overline{\beta}(i(p,t))[a] = \widehat{W}((p,a),(t,\beta))$. Similarly we get $r(e^\bullet)(p)[a]=\widehat{W}((t,\beta),(p,a))$.
   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 $|\{b \in {}^\bullet e \mid r(b)=\hat{p}\}| = \widehat{W}(\hat{p},\hat{t})$ and $|\{b \in e^\bullet \mid r(b)=\hat{p}\}| = \widehat{W}(\hat{t},\hat{p})$.
   This is condition 2 of Def. 29.
   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 very useful in the verification of distributed algorithms.

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 [20] and Reisig [15], algebraic system nets as proposed in this paper show some fundamental differences:

1. There are no flexible arcs in [20, 15].
2. Reisig [15] uses algebraic specifications [6] 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 [15] 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 form Petri net concepts such as place invariants to temporal properties (cf. [16, 11, 21, 10]).
4. Reisig [15] introduces a firing 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 [1] of the unfolding.

Acknowledgements We thank Wolfgang Reisig and Karsten Schmidt for helpful suggestions and comments.

References

1. E. Best and C. Fernández. *Nonsequential Processes*, volume 13 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1988.
2. J. Billington. Extending coloured Petri nets. Technical Report 148, University of Cambridge, Computer Laboratory, Oct. 1988.
3. J. Billington. *Extensions to Coloured Petri Nets and their Application to Protocols*. Technical report no. 222, University of Cambridge, May 1991.
4. M. Broy. On the design and verification of a simple distributed spanning tree algorithm. SFB-Bericht 342/24/90 A, Technische Universtität München, Dec. 1990.
5. J. Desel, K.-P. Neuendorf, and M.-D. Radola. Proving nonreachability by modulo-invariants. *Theoretical Comput. Sci.*, 153:49–64, 1996.
6. H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specifications 1, Equations and Initial Semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.
7. U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Information and Control*, 57:125–147, 1983.
8. K. Jensen. *Coloured Petri Nets. Volume 2: Analysis Methods*. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1995.

9. E. Kindler and W. Reisig. Algebraic system nets for modelling distributed algorithms. *Petri Net Newsletter*, 51:16–31, Dec. 1996.
10. E. Kindler and W. Reisig. Verification of distributed algorithms with algebraic Petri nets. In C. Freksa, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science: Potential – Theory – Cognition*, volume 1337 of *LNCS*, pages 261–270. Springer-Verlag, 1997.
11. E. Kindler, W. Reisig, H. Völzer, and R. Walter. Petri net based verification of distributed algorithms: An example. Informatik-Berichte 63, Humboldt-Universität zu Berlin, May 1996. to appear in Formal Aspects of Computing, 1997.
12. G. Memmi and G. Roucairol. Linear algebra in net theory. In W. Brauer, editor, *Net Theory and Applications*, volume 84 of *LNCS*, pages 213–223. Springer-Verlag, Oct. 1979.
13. J. L. Peterson. *Petri Net Theory And The Modeling of Systems.* Prentice-Hall, 1981.
14. W. Reisig. *Petri Nets*, volume 4 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, 1985.
15. W. Reisig. Petri nets and algebraic specifications. *Theoretical Comput. Sci.*, 80:1–34, May 1991.
16. W. Reisig. Petri net models of distributed algorithms. In J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, volume 1000 of *LNCS*, pages 441–454. Springer-Verlag, 1995.
17. K. Schmidt. Verification of siphons and traps for algebraic Petri nets. In P. Azéma and G. Balbo, editors, *Application and Theory of Petri Nets 1997, Internat. Conference, Proceedings*, volume 1248 of *LNCS*, pages 427–446. Springer-Verlag, June 1997.
18. E. Smith and W. Reisig. The semantics of a net is a net, an exercise in general net theory. In K. Voss, H. Genrich, and G. Rozenberg, editors, *Concurrency and Nets.* Springer-Verlag, 1987.
19. R. Valk. Bridging the gap between place- and Floyd-invariants with applications to preemtive scheduling. In M. A. Marsan, editor, *Application and Theory of Petri Nets 1993, International Conference, Proceedings*, volume 691 of *LNCS*, pages 431–452. Springer-Verlag, June 1993.
20. J. Vautherin. Parallel systems specifications with coloured Petri nets and algebraic specifications. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 266 of *LNCS*, pages 293–308. Springer-Verlag, 1987.
21. R. Walter, H. Völzer, T. Vesper, W. Reisig, E. Kindler, J. Freiheit, and J. Desel. Memorandum: Petrinetzmodelle zur Verifikation verteilter Algorithmen. Informatik-Bericht 67, Humboldt-Universität zu Berlin, July 1996.