On the Computation of Place Invariants for Algebraic Petri Nets

Karsten Schmidt
Humboldt–Universität zu Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin, Germany
e-mail: kschmidt@informatik.hu-berlin.de

Abstract

The paper is concerned with the computation of a generator set for the space of all place invariants for a given algebraic net. We will show that the problem can be divided into two major steps. First we trace back the problem to a set of equations between terms. Then we combine the solutions of these equations to obtain the solution of the original problem. For both steps we present a solution for a restricted class of algebraic nets, where the algebraic specification contains no equations and at most unary operation symbols.

1 Introduction

Computer aided analysis of concurrent systems is a difficult problem. For Petri nets with individual token most of the usual analysis problems become undecidable as soon as there are infinitely many possible inscriptions for a token. Such nets appear for instance as models of concurrent algorithms. Nevertheless there are methods for the analysis of such models, among them the invariant method, which has been established for almost every high–level net calculus. With the help of this calculus several system properties can be expressed and automatically verified. This verification yields a set of "user intended" place invariants. Additionally it is possible to use place invariants to prove the non–reachability of many markings, since the application of a place invariant yields the same value for all *reachable* markings. This non–reachability test can be done automatically. The more independent place invariants are involved in this test, the more states can be classified to be non–reachable. Therefore it would be useful to have the possibility to compute a generator set of *all* place invariants of a high–level Petri net, including possibly some which are not user intended. In the sequel we will discuss an approach how the problem of computing a generator set of all place invariants can be solved for algebraic Petri nets. The idea is to trace back the problem to a set of equations between terms which appear in the incidence matrix of the net. We present a solution of these equations for a restricted class of Petri nets. We assume an equation free specification and the absence of operation symbols with an arity greater than one. This restriction is very strong but covers a lot of interesting nets, among them the famous philosophers nets and all kinds of counter automata (including their composition via different kinds of synchronization). After having solved the above mentioned system of equations its solutions must be combined to obtain

invariants. Again we present a solution for the mentioned net class. Finally we discuss the problems which appear when the restrictions are weakened.

2 Basic Definitions

The definitions are based on [EM85] and [Re91].

Definition 2.1 (Specifications) *A* **signature** $\Sigma = [S, \Omega]$ *consists of a set* S *of* **sorts** *and a family* $\Omega = \{\Omega_{w,s}\}_{w \in S^*, s \in S}$ *of* **operation symbols**. *For* e *being the empty word,* $\Omega_{e,s}$ *is the set of* **constant symbols** *of sort* s. *A set of* Σ**–variables** *is a family* $X = \{X_s\}_{s \in S}$ *of* **variables**. *The set* $T_{\Omega,s}(X)$ *of* (Ω, X)**–terms of sort** s *is inductively defined by 1.* $X_s \cup \Omega_{e,s} \subseteq T_{\Omega,s}(X)$ *and 2. for* $\omega \in \Omega_{s_1 \cdots s_n, s}$ *and* $T_i \in T_{\Omega,s_i}(X)$, $\omega(T_1, \cdots, T_n) \in T_{\Omega,s}(X)$. *The set* $T_{\Omega,s} := T_{\Omega,s}(\emptyset)$ *contains the* **ground terms of sort** s, $T_\Omega(X) := \bigcup_{s \in S} T_{\Omega,s}(X)$ *is the set of* Σ**–terms over** X, *and* $T_\Omega := T_\Omega(\emptyset)$ *is the set of* Σ**–ground terms**. *A* Σ**–equation** *of sort* s *over* X *is a pair* $[L, R]$ *of terms* $L, R \in T_{\Omega,s}(X)$. *A* **specification** $D = [\Sigma, E]$ *consists of a signature* Σ *and a set* E *of* Σ*–equations.*

Definition 2.2 (Algebras) *A* Σ**–algebra** $A = [S_A, \Omega_A]$ *consists of a family* $S_A = \{s_A\}_{s \in S}$ *of* **domains** *and a set* $\Omega_A = \{\omega_A \mid \omega \in \Omega\}$ *of* **operations**, *where* $\omega_A : s_{1A} \times \cdots \times s_{nA} \to s_A$ *for* $\omega \in \Omega_{s_1 \cdots s_n, s}$. *The elements* ω_A *for* $\omega \in \Omega_{e,s}$ *can be identified with elements of* s_A. *An* **assignment** *is a family* $\alpha = \{\alpha_s\}_{s \in S}$ *of mappings* $\alpha_s : X_s \to s_A$. *An* **evaluation** *according to an assignment* α *is a family of mappings* $\{\alpha_s^\#\}_{s \in S}$ *with* $\alpha_s^\# : T_{\Omega,s}(X) \to s_A$ *which is defined inductively by 1.* $\alpha_s^\#(x) := \alpha_s(x)$ *for* $x \in X_s$, *and 2.* $\alpha_s^\#(\omega(T_1, \cdots, T_n)) := \omega_A(\alpha_{s_1}^\#(T_1), \cdots, \alpha_{s_n}^\#(T_n))$ *for* $\omega \in \Omega_{s_1 \cdots s_n, s}$. *For ground terms* $T \in T_{\Omega,s}$ *we define the* **value** *of* T *in* A $\#_A(T) := \alpha_s^\#(T)$ *for an arbitrary assignment* α *(the value is actually not dependent on* α, *since ground terms do not contain variables). A* Σ*–equation* $[L, R]$ *is* **valid** *in a* Σ*–algebra* A *iff for all assignments* α, $\alpha^\#(L) = \alpha^\#(R)$. *For a specification* $D = [\Sigma, E]$ *the* Σ*–algebra* A *is a* D**–algebra** *(or a* **model** *of* D*) iff all the equations in* E *are valid in* A.

Definition 2.3 (Substitutions) *Let* X *and* Y *be two sets of* Σ*–variables. A* **substitution** σ *over* X *is an assignment* $\sigma : X \to T_\Omega(Y)$, *(*$X_s \to T_{\Omega,s}(Y)$*). A* **ground substitution** *is a substitution* $\sigma : X \to T_\Omega$. *An injective substitution* $\sigma : X \to Y$ *is called* **renaming**. *For a term* T *and a substitution* σ *the term* $T\sigma$ *results from simultaneously replacing the variables in* T *by their corresponding* σ*–values. The application of a substitution* σ *with* $\sigma(x) = T$ *and* $\sigma(y) = y$ *on all variables except* x *to a term* T' *is written* $T'_{<x \leftarrow T>}$.

Definition 2.4 (Term Equivalence) *Two terms* T_1 *and* T_2 *are* **equivalent** *according to a specification* $D = [\Sigma, E]$ *(*$T_1 \equiv_E T_2$*) iff for all* D*–algebras* A *and all assignments* α *in* A, $\alpha_A^\#(T_1) = \alpha_A^\#(T_2)$.

$\equiv_E$ is an equivalence relation on $T_\Omega(X)$. It is actually a congruence relation, i.e. $T_1 \equiv_E T_2$ implies $T_1\sigma \equiv_E T_2\sigma$ for arbitrary substitutions σ. With $[T]_E$ we denote the equivalence class of the term T according to the relation $\equiv_E$.

Definition 2.5 (Initial Algebra) *Let* $D = [\Sigma, E]$ *be a specification. The* **initial algebra** I *of* D *consists of the domains* $s_I := \{[T]_E \mid T \in T_{\Omega,s}\}$ *and the operations* ω_I *with* $\omega_I([T_1]_E, \cdots, [T_n]_E) := [\omega(T_1, \cdots, T_n)]_E$.

Due to the properties of the relation $\equiv_E$ the initial algebra is a model of D. Furthermore it satisfies the "no junk" property (every element is represented by a ground term) and the "no confusion" property (there are no valid equations except those implied by E). Though there are several models for a specification and it is challenging to obtain results which are valid for several models, we will consider exclusively initial algebras in the sequel.

Definition 2.6 (Multisets) *For a set M, a* **multiset** *over M is a mapping from M into the integer numbers. A multiset is* **semipositive** *iff all the values are greater or equal 0. A multiset is* **finite** *iff it has finite support. The* **empty** *multiset over M, denoted by ϑ_M, assigns 0 to every element of M. For an element $m \in M$, the multiset $\underline{m}$ assigns 1 to m and 0 to every other $m' \in M$. The multisets $\mu_1+\mu_2$ and $\mu_1-\mu_2$ are defined by $(\mu_1+\mu_2)(m) := \mu_1(m)+\mu_2(m)$ and $(\mu_1-\mu_2)(m) := \mu_1(m)-\mu_2(m)$. This way every finite multiset can be represented as a* **formal sum** *of the $\underline{m}(m \in M)$. In such formal sums we usually write m instead of $\underline{m}$. A multiset μ_1 is* **less or equal** *to μ_2 iff for all $m \in M$, $\mu_1(m) \leq \mu_2(m)$.* We write $2 \cdot a + 3 \cdot b$ for $a + a + b + b + b$.

Definition 2.7 (Algebraic Petri Nets) *$AN = [D; \mathbf{P}, \mathbf{T}, \mathbf{F}; \psi, \xi, \lambda; m_0]$ is an* **algebraic Petri net** *iff (1) $D = [\Sigma, E]$ is a specification with $\Sigma = [S, \Omega]$; (2) $[\mathbf{P}, \mathbf{T}, \mathbf{F}]$ is a* **net**, *i.e.* **P** *and* **T** *are finite and disjoint sets called* **places** *and* **transitions**, *respectively, and* **F** *is a relation $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$, the elements of which are called* **arcs**; *(3) ψ is a* **sort assignment** *$\psi : \mathbf{P} \to S$; (4) ξ assigns a finite set of Σ–variables $\xi(t)$ to each transition $t \in \mathbf{T}$; (5) λ is the* **arc inscription** *such that for $f = [p, t]$ or $f = [t, p]$ in* **F**, *$\lambda(f)$ is a finite multiterm over $T_{\Omega,\psi(p)}(\xi(t))$; (6) m_0 is a* **marking**, *i.e. it assigns a finite multiterm over $T_{\Omega,\psi(p)}$ to every $p \in \mathbf{P}$. m_0 is called the* **initial marking**.

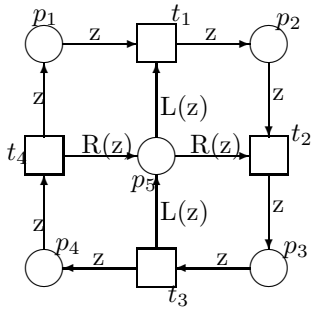

Figure 1:

Example. Figure 1 shows an algebraic Petri net. It is based on a specification where $S = \{phil, fork\}$, $\Omega_{\epsilon,phil} = \{a, b, c, d, e\}$, $\Omega_{phil,phil} = \{succ\}$, $\Omega_{phil,fork} = \{L, R\}$, $X_{phil} = \{x\}$ and $E = \{succ(a) = b, succ(b) = c, succ(c) = d, succ(d) = e, succ^5(x) = x, R(x) = L(succ(x))\}$. Furthermore assume, that $\psi(p_1) = \cdots = \psi(p_4) = phil$, $\psi(p_5) = fork$, $\xi(t_1) = \cdots = \xi(t_4) = \{z\}$ and λ is defined as depicted in the figure (for instance: $\lambda([p_5, t_2]) = R(z); \lambda([t_4, p_1]) = z$). The initial marking shall be $a + \cdots + e$ on p_1 and $L(a) + \cdots + L(e)$ on p_5.

For $f \notin \mathbf{F}$ we define $\lambda(f) := \vartheta$. With t^- and t^+ we denote the $\mathbf{P}$–indexed vectors defined by $t^-(p) := \lambda([p, t])$ and $t^+(p) := \lambda([t, p])$, respectively. We

write Δt for $t^+ - t^-$. The $(\mathbf{P} \times \mathbf{T})$–matrix where the column belonging to t is Δt is called the **incidence matrix** of the algebraic net in question. It is possible to interpret an algebraic Petri net according to an arbitrary model of the specification D. The result is a colored net. This way all the behavioral aspects of an algebraic net can be traced back to colored nets. We define the transition rule of an algebraic net for its initial algebra model only.

Definition 2.8 (Transition Rule) *Any ground substitution β of $\xi(t)$ is an* **occurrence mode** *of transition $t \in \mathbf{T}$. A transition $t \in \mathbf{T}$ is* **enabled** *in an occurrence mode β at a marking m iff for all $p \in \mathbf{P}$ with $[p,t] \in \mathbf{F}$, $[\lambda([p,t])\beta]_E \leq [m(p)]_E$. If t is enabled in β at m, then t may* **fire** *yielding the marking m', where for all $p \in \mathbf{P}$, $m'(p) = m(p) - \lambda([p,t])\beta + \lambda([t,p])\beta$. We write $m \xrightarrow{t,\beta} m'$ in this case. The set $R_{AN}(m^*)$ of markings* **reachable** *from a target marking m^* is the smallest set of markings, which contains m^* and if $m \in R_{AN}(m^*)$ and $m \xrightarrow{t,\beta} m'$ for some occurrence mode β, then there is a marking in $R_{AN}(m^*)$ which is componentwise equivalent to m' w. r. t. $\equiv_E$.*

3 Place Invariants

Place invariants provide weight functions for every place in such a way, that the weighted sum of all tokens remains invariant under transition occurrences. For algebraic nets (as well as for coloured nets, conf. [Je81b]) the weights are multisets of a color domain. Syntactically the weight functions are expressed as multiterms with one distinguished variable which we will always call x or "the argument variable" in the sequel. Other variables may occur in the multiterms. These variables will be written y and called "parameter variables". For the place invariant approach it is of major importance to distinguish clearly between these two completely different kinds of variables. The value of the weight function F for a given argument T is the multiterm which results from F by replacing every occurrence of x in F by T (this is the special task of the variable x which makes it necessary to distinguish it from the remaining variables). The formalism has to pay attention to the different sorts.

Definition 3.1 *Let $\Sigma = [S, \Omega]$ be a signature with $s_1, s_2 \in S$, Y a family of Σ–variables and x_s $(s \in S)$ a variable of sort s with $x_s \notin Y$. The set of* **weight terms from** s_1 **to** s_2 *is the set $T_{\Omega,s_2}(Y \cup \{x_{s_1}\})$ of terms of sort s_2 with the* **argument variable** x_{s_1} *and* **parameter variables** *from Y. The set of* **weight functions from** s_1 **to** s_2 $W_{s_1,s_2}(Y)$ *is the set of all finite multiterms over $T_{\Omega,s_2}(Y \cup \{x_{s_1}\})$, i.e. all finite multisets of weight terms from s_1 to s_2.*

Now we define the application of a weight term W to a term T (written $W \cdot T$).

Definition 3.2 *For $W \in W_{s_1,s_2}(Y)$ and $T \in T_{\Omega,s_1}(Y)$, $W \cdot T := W_{<x_{s_1} \leftarrow T>}$.*

Consequently, a weight term represents a function which assigns an element of sort s_2 (namely $W_{<x_{s_1} \leftarrow T>}$) to every element of sort s_1 (represented by T). The parameter variables are free parameters of the invariant. The definition allows weight terms where x_{s_1} does not appear. In this case the value of the weight terms is independent from the argument, that is a constant function. The application of a weight function F to a multiterm M is defined as the

linear extension of the above product. Furthermore the product can be linear extended to a product between vectors of weight functions and vectors of multiterms. Usually the sort of the argument variable x_{s_1} is clear from the context and therefore we skip the sort index in the remaining part of the paper.

Definition 3.3 ([Re91]) *Let $AN = [S, \Omega, E; \mathbf{P}, \mathbf{T}, \mathbf{F}; \psi, \xi, \lambda, m_0]$ be an algebraic net and Y a family of Σ–variables being disjoint to all $\xi(t), t \in \mathbf{T}$. A* **place invariant** *of sort s is a $\mathbf{P}$–indexed vector I where $I(p)$ is a weight function from $\psi(p)$ to s such that for every marking m' reachable from a marking m it holds $I \cdot m \equiv_E I \cdot m'$.*

Example. The vector $\left(\begin{array}{c|l} p_2 & L(x) \\ p_3 & L(x) + R(x) \\ p_4 & R(x) \\ p_5 & x \end{array}\right)^*$ is a place invariant for the dining philosopher's model. The weight of the initial marking is $\vartheta_{<x \leftarrow (a+\cdots+e)>} + x_{<x \leftarrow (L(a)+\cdots+L(e))>} = \vartheta + L(a) + \cdots + L(e) = L(a) + \cdots + L(e)$ The weight of a marking with some token x and $succ(x)$ on p_3 would contain $2 \cdot L(succ(x))$. Hence the weight of such a marking differs from the weight of the initial marking where all elements appear only once and therefore it cannot be reachable.

Of course the arguments of the weight function for a place p are token values on p, that is terms of sort $\psi(p)$. The result of the application of a place invariant I to a marking m is a multiterm of sort s — the weight of m with respect to I. A vector I of weight functions is a place invariant iff transition occurrences do not change the weights of markings. The following theorem characterizes place invariants (C^T is the transposed of the incidence matrix C).

Theorem 1 ([Re91]) *I is a place invariant of an algebraic net having the incidence matrix C iff $C^T \cdot I \equiv_E \underline{\vartheta}$.* □

The weight of markings remains invariant under transition occurrences iff for every transition $t \in \mathbf{T}$ the *change* of the weight caused by t is the empty multiterm. The changes of weights by single transition occurrences are expressed by a product $C^T \cdot I$ (the linear extension of the above defined product between multiterms and weight functions).

Example. For the above invariant it holds $C^T \cdot I =$

$$= \left(\begin{array}{c|cccc} & t_1 & t_2 & t_3 & t_4 \\ \hline p_1 & -z & & & z \\ p_2 & z & -z & & \\ p_3 & & z & -z & \\ p_4 & & & z & -z \\ p_5 & -L(z) & -R(z) & L(z) & R(z) \end{array}\right)^{\mathrm{T}} \cdot \left(\begin{array}{c|l} p_1 & \\ p_2 & L(x) \\ p_3 & L(x) + R(x) \\ p_4 & R(x) \\ p_5 & x \end{array}\right)$$

$$= \left(\begin{array}{c|l} t_1 & L(x) \cdot z - x \cdot L(z) \\ t_2 & -L(x) \cdot z + (L(x) \\ & +R(x)) \cdot z - x \cdot R(z) \\ t_3 & -(L(x) + R(x)) \cdot z \\ & +R(x \cdot z + x \cdot L(z) \\ t_4 & -R(x) \cdot z + x \cdot R(z) \end{array}\right) = \left(\begin{array}{c|l} t_1 & L(z) - L(z) \\ t_2 & -L(z) + L(z) + \\ & R(z) - R(z) \\ t_3 & -L(z) - R(z) + \\ & R(z) + L(z) \\ t_4 & -R(z) + R(z) \end{array}\right) = \underline{\vartheta}$$

*Unmentioned components (here: p_1) are assumed to be empty (or ϑ, resp.)

Consequently the problem we have to consider in the sequel is to find a generator set for all solutions of the system of equations $C^T \cdot I \equiv_E \underline{\vartheta}$, where I is the vector of unknowns. For the computation of generator sets it is important to have in mind the operations which preserve invariance. Among these operations are obviously linear combinations.

Theorem 2 ([Re91]) *With I_1 and I_2, also $I_1 + I_2$ and $I_1 - I_2$ are place invariants.* □

Additionally we exploit the congruence properties of $\equiv_E$.

Lemma 3 *Let W_1 and W_2 be two weight terms from a sort s_1 to a sort s_2 with a set Y of parameter variables. If for some terms T_1 and T_2 taken from $T_\Omega(X)$ (X and Y are assumed to be disjoint) it holds $W_1 \cdot T_1 \equiv_E W_2 \cdot T_2$, then it holds also (1.) $(W_1\sigma)\cdot T_1 \equiv_E (W_2\sigma)\cdot T_2$ for arbitrary substitutions σ over Y (without substituting the argument variable x!) and (2.) $(W \cdot W_1) \cdot T_1 \equiv_E (W \cdot W_2) \cdot T_2$ for arbitrary weight terms W from s_2 to some sort s_3.*

Proof. Follows immediately from the congruence properties of $\equiv_E$ and the definition of the product between terms and weight functions. □

This lemma immediately leads to the observation that instantiation of parameter variables and concatenation of weight functions preserve invariance.

Theorem 4 *Let Y be a family of Σ–variables, s a sort and I an invariant where $I(p)$ is a weight function from $\psi(p)$ to s. Let $\sigma : Y \to T_\Omega(Y)$ be a substitution of the parameter variables in I. Then $I\sigma$ defined by $I\sigma(p) := I(p)\sigma$ is a place invariant.* (Straightforward from Lemma 3) □

Theorem 5 *Let I be a place invariant of sort s_1 and W a weight function from sort s_1 to sort s_2. Then $W \cdot I$ defined by $(W \cdot I)(p) := W \cdot I(p)$ is a place invariant of sort s_2.* (Straightforward from Lemma 3.) □

4 Incidence Matrix and Term Equations

The only input for the computation of place invariants is the incidence matrix. We have to exploit this matrix to obtain conditions which distinguish invariants from place vectors of multiterms which are no invariants. For this purpose we regard the process of verifying a given invariant. Let I be an appropriate **P**–indexed vector of weight functions, that is a **P**–indexed vector which satisfies all syntactical conditions for a place invariant. According to Theorem 1 we build $C^T \cdot I$. The result is a **T**–indexed vector of multiterms. Thereby *every* term appearing in a component of this vector results from applying a weight term from the **P**–indexed vector to a term of the incidence matrix. Its multiplicity is the product of multiplicities of the weight term and the matrix term. Then we have to check whether the resulting vector is equivalent to the vector of empty multiterms. Therefore we have to build the equivalence classes of terms in every component of the **T**–indexed vector and to check, whether the sum of the multiplicities in every resulting equivalence class is 0.

Example. Consider the philosopher's net and the invariant considered in the previous section. The **T**–indexed vector resulting from the product of C^T

and I is $\left(\begin{array}{c|l} t_1 & L(z) - L(z) \\ t_2 & -L(z) + L(z) + R(z) - R(z) \\ t_3 & -L(z) - R(z) + R(z) + L(z) \\ t_4 & -R(z) + R(z) \end{array}\right)$. The equivalence classes of terms in the second component of this vector are $[L(z)]$ and $[R(z)]$ while in the first component there is only the equivalence class $[L(z)]$. Obviously the sum of all multiplicities of terms belonging to one and the same equivalence class is always zero.

In this example the detection of equivalence classes is very simple. Usually there are equivalent terms which are syntactically different (due to the specified set of equations). From this observation we learn, that the application of a weight term to a matrix term results in a term which is equivalent to the results of the application of other weight terms to matrix terms. In this sense we can consider the weight terms contained in the invariant to be solutions of sets of equations like $X_1 \cdot T_1 \equiv_E \cdots \equiv_E X_n \cdot T_n$ where $X_1, \cdots, X_n$ are the unknown weight terms, "$\cdot$" is the product between weight terms and matrix terms which has been defined in the previous section and $T_1, \cdots, T_n$ are terms appearing in one and the same column of the incidence matrix (otherwise the results of applying weight terms to them would not appear in the same component of $C^T \cdot I$).

Example. The weight terms $X_1 := L(x)$ and $X_2 := x$, occurring in the components p_2 and p_5, resp. in the running example, are solutions of the equation $X_1 \cdot z \equiv_E X_2 \cdot L(z)$. This equation reflects the equivalence class $[L(z)]$ in the component t_1 of $C^T \cdot I$ in the above example where one of the two "$L(z)$" origins from $L(x) \cdot z$ and the other "$L(z)$" from $x \cdot L(z)$.

The first major idea for computing a generator set of all place invariants for a given net is to find "most general solutions" for the above kind of equations for all sets of terms appearing in the same column of the incidence matrix and to construct invariants based on those most general solutions.

In the sequel we discuss how to find the most general solutions of equations for weight terms in the case, that the specification consists only of constant symbols and unary operation symbols and the set of equations is empty. For reasons of simplicity we consider only single equations (i.e. $X_1 \cdot T_1 \equiv_E X_2 \cdot T_2$) but the results can be straightforward generalized to n unknowns. We introduce two notations for operating on terms which can be built in the restricted kind of specifications. Let $T_1 = \omega_m(\omega_{m-1}(\cdots(\omega_1(r))\cdots))$ and $T_2 = \omega'_n(\omega'_{n-1}(\cdots(\omega'_1(r'))\cdots))$ be terms where r and r' are variables or constant symbols. We say $T_1 \sqsubseteq T_2$ iff $m \leq n$, $r = r'$ and for all i $(1 \leq i \leq m)$ it holds $\omega_i = \omega'_i$. If $T_1 \sqsubseteq T_2$ the term $T_2 \setminus T_1$ is defined as $\omega'_n(\cdots(\omega'_{m+1}(x))\cdots)$. Thereby $T_2 \setminus T_1$ is always considered to be a weight term and its variable x is the argument variable of the sort of T_1. Thus for terms T_1 and T_2 with $T_1 \sqsubseteq T_2$ we have always the relation $(T_2 \setminus T_1) \cdot T_1 = T_2$ (= is the syntactical identity of terms). These operations are sufficient to construct a set of most general solutions of the above kind of equations.

Definition 4.1 *The* **set of most general solutions** MGS *for the equation* $X_1 \cdot T_1 \equiv_\emptyset X_2 \cdot T_2$ *is defined as follows:*
1. $[X_1 = y, X_2 = y] \in MGS$ *(y is an arbitrary* parameter *variable);*
2. If T_1 *is a ground term, then* $[X_1 = x, X_2 = T_1] \in MGS$*;*
3. If T_2 *is a ground term, then* $[X_1 = T_2, X_2 = x] \in MGS$*;*
4. If $T_1 \sqsubseteq T_2$*, then* $[X_1 = T_2 \setminus T_1, X_2 = x] \in MGS$ *;*

5. If $T_2 \sqsubseteq T_1$, then $[X_1 = x, X_2 = T_1 \setminus T_2] \in MGS$;
6. MGS contains no elements except those according to the first five items.

Theorem 6 *All elements of MGS are solutions of the given equation.* □

Due to Lemma 3 instantiating the variable y and concatenating the solutions with other weight terms lead to solutions again. More interesting is the following justification for the name "most general solutions".

Theorem 7 *Every solution of the equation $X_1 \cdot T_1 \equiv_\emptyset X_2 \cdot T_2$ can be obtained from a solution in the corresponding MGS by instantiating the parameter variable y or by concatenating a solution with an arbitrary weight term.*

Proof. Let $[X_1 = W_1, X_2 = W_2]$ be a solution of the given equation, that is $W_1 \cdot T_1 \equiv_\emptyset W_2 \cdot T_2$. Assume first that both W_1 and W_2 are terms containing unary operation symbols, say $W_1 = \omega_1(W_1')$ and $W_2 = \omega_2(W_2')$. We obtain $\omega_1(W_1')_{<x \leftarrow T_1>} \equiv_\emptyset \omega_2(W_2')_{<x \leftarrow T_2>}$ leading to $\omega_1(W'_{1<x \leftarrow T_1>}) \equiv_\emptyset \omega_2(W'_{2<x \leftarrow T_2>})$. From this equation we may conclude (1.) $\omega_1 = \omega_2$ (since $\equiv_\emptyset$ is the syntactical identity on terms); (2.) $W'_{1<x \leftarrow T_1>} \equiv_\emptyset W'_{2<x \leftarrow T_2>}$, that is $[X_1 = W_1', X_2 = W_2']$ is a solution of the given equation. (3.) The solution $[X_1 = W_1, X_2 = W_2]$ can be obtained from the solution $[X_1 = W_1', X_2 = W_2']$ by concatenating it with $\omega_1(x)$. Hence every solution where both weight function contain unary operation symbols can be obtained by concatenation from a solution where at least one of the involved weight functions does not contain unary operation symbols. It remains to show that these solutions are generated by MGS. Since all the considerations for W_1 and W_2 are symmetric, we may assume, that W_1 is the weight function which does not contain unary operation symbols. Due to the restricted set of operation symbols there remain three cases for W_1.

Case 1: $W_1 = x$ (the argument variable). In this case it holds $W_1 \cdot T_1 = T_1$. On the other hand W_2 does or does not contain the variable x, too. If it does, the term T_2 appears at the bottom of $W_2 \cdot T_2$. Therefore it holds $T_2 \sqsubseteq T_1$ and due to the syntactical identity of T_1 and $W_2 \cdot T_2$ we obtain $W_2 = T_1 \setminus T_2$. This is covered by item 4 (or 5, resp.) of Def. 4.1. If W_2 does not contain x, then it holds $W_2 \cdot T_2 = W_2$ leading to $T_1 \equiv_\emptyset W_2$. Since the variable sets of weight functions are disjoint from the variable sets of terms from the incidence matrix, T_1 must be a ground term and it holds $W_2 = T_1$ (syntactically) due to the absence of equations. This is covered by item 2 (and 3, resp.) of Def. 4.1.

Case 2: $W_1 = y$ (a parameter variable). Then $W_1 \cdot T_1 = y$. Therefore, since y must appear in $W_2 \cdot T_2$, too but cannot appear in T_2, it must appear in W_2. Thus we have $W_2 \cdot T_2 = W_2$ and $y = W_1 = W_2 \cdot T_2 = W_2$, that is $W_2 = y$. This is covered by item 1 of Def. 4.1.

Case 3: $W_1 = \omega$ for some constant symbol ω. Then we have $W_1 \cdot T_1 = \omega \equiv_\emptyset W_2 \cdot T_2$. Since $W_2 \cdot T_2$ contains at least the operation symbols of W_2, W_2 cannot contain unary operation symbols (else there would be no way to obtain syntactical identity of ω and $W_2 \cdot T_2$). Therefore it remains to consider the case that W_2 consists of a constant symbol only, since all other cases can be covered by one of the first to cases through swapping W_1 and W_2. If W_2 is a constant symbol, then we have $W_2 \cdot T_2 = W_2$ leading to $W_2 = \omega$. This solution thus can

be obtained from the solution $[X_1 = y, X_2 = y]$ by instantiating the variable y with ω. □

At this stage we will demonstrate the problems which appear when the above restrictions are weakened. The problem caused by the introduction of equations seems to be roughly the same as for the well–known technique of unification modulo a set of equations. That is, the existence of solutions is probably undecidable and the number of solutions (finitely many or infinitely many independent solutions) will probably depend on the design of the specification. We will concentrate on the second restriction. Assume there is a binary operation symbol f and there are at least two different ground terms T_1 and T_2. Then there are infinitely many solutions for the equation $X_1 \cdot T_1 \equiv_\emptyset X_2 \cdot T_2$. Among others there are the following solutions:
$[X_1 = f(x, T_2), X_2 = f(T_1, x)]$
$[X_1 = f(x, f(y_1, T_2)), X_2 = f(T_1, f(y_1, x))]$
$[X_1 = f(x, f(y_1, f(y_2, T_2))), X_2 = f(T_1, f(y_1, f(y_2, x)))] \cdots$

All these solutions are independent and cannot be covered by a finite set of solutions. Therefore there is no suitable way to weaken the restriction unless a) a complete different way of computing the invariants is found, b) the concept of invariants is generalized in a way, that all the solutions can be represented finitely or c) we have some additional information on how many of these solutions are sufficient to get enough invariants for a strong non–reachability test. Up to now we have not considered any of these possibilities and so we must leave their consideration to the future. In the next section we describe how to build invariants from the solutions considered here.

5 From Term Equations to Invariants

In the previous section we considered term equations in order to characterize the equivalence classes of terms in the components of the **T**–indexed vector $C^T \cdot I$ and to constrain the weight terms from which invariants can be composed.

(a) (b)

Figure 2:

Example. Consider the net in Figure 2(a). The only interesting details concerning the specification are that c and d are constants while z_1 and z_2 are variables. The incidence matrix of this net is $\begin{pmatrix} -z_1 & \\ c & -d \\ & z_2 \end{pmatrix}$. Thereby places and transitions are assumed to be numbered consecutively from left to right in the figure. The only reasonable equations obtainable from the columns of this matrix are $X_1 \cdot c \equiv_\emptyset X_2 \cdot z_1$ for t_1 and $X_1 \cdot z_2 \equiv_\emptyset X_2 \cdot d$ for t_2. The MGS are $\{[X_1 = y, X_2 = y], [X_1 = x, X_2 = c]\}$ for the first and $\{[X_1 = y, X_2 = y], [X_1 = d, X_2 = x]\}$ for the second equation. But it is impossible to build invariants from c (for p_1 — z_1 belongs to an arc from p_1) and x (for p_2 — c belongs to an arc to p_2) only. In the component t_2 of $C^T \cdot I$ there would appear

the term d $(= x \cdot d)$ with a non–zero multiplicity. Nevertheless there is an invariant for this net providing an equivalence class which is characterized by this equation and the considered solution, namely $(c, x, d)^T$. For this invariant it holds $C^T \cdot I = (c - c, -d + d)^T$. The equivalence class $[c]$ in the first component corresponds to the equation $X_1 \cdot z_1 \equiv_\emptyset X_2 \cdot c$ with the solution $[X_1 = c, X_2 = x]$, just the considered one. The crucial point is that in this invariant for the product of the weight term x in the second component and the matrix term d there is another equivalence class, namely $[d]$ in the second component of $C^T \cdot I$. That is, the weight term x is shared between the solutions of two different equations, namely as X_2 in the considered solution of $X_1 \cdot z_1 \equiv_E X_2 \cdot c$ and as X_1 in the solution $[X_1 = x, X_2 = d]$ of the equation $X_1 \cdot d \equiv_\emptyset X_2 \cdot z_2$.

A closer observation of how we multiply C^T and I shows that it is unavoidable to share weight terms in solutions of different term equations, since *every weight term must be applied to every term of the corresponding row of the incidence matrix and the results of this application usually appear in different equivalence classes.* Remember that the term equations correspond to the equivalence classes in $C^T \cdot I$. This leads us to a stronger relation between the weight terms which occur in a place invariant. We define a special concept for this relation.

Definition 5.1 *Let U be a set of ordered pairs $[Q, S]$ where Q is an equation of the kind $X_1 \cdot T_1 \equiv_E \cdots \equiv_E X_n \cdot T_n$ with $T_1, \cdots, T_n$ occurring in one and the same column of the incidence matrix C and S is a solution of Q. U is called* **complete system of solutions** *of the system $\{Q \mid \exists S : [Q, S] \in U\}$ iff for every weight term W appearing in a solution S of a $[Q, S] \in U$ corresponding to the part $X_i \equiv_E T_i$ of Q and for every term T which appears in the same row of C as T_i there is a $[Q', S'] \in U$ where for some j the equation $X_j \equiv_E T_j$ is part of Q' and $X_j = W$ is the corresponding solution in S'.*

Corollary 8 *Weight terms occurring in a place invariant always appear in a complete system of solutions of a set of equations where every equation contains only terms of one and the same column of C.*

From a complete system of solutions the component of I where a weight term W appears is determined by the rows for which the completeness condition has to be established.

Example. Consider the net in Figure 2(a). $U = \{[X_1 \cdot z_1 \equiv_\emptyset X_2 \cdot c, [X_1 = c, X_2 = x]], [X_1 \cdot d \equiv_\emptyset X_2 \cdot z_2, [X_1 = x, X_2 = d]]\}$ is a complete system of solutions since c appears corresponding to every term of the first row of C (only z_1), x appears corresponding to every term of the second row (for c in the first and d in the second pair of U) and d appears corresponding to the only term z_2 of the last row of c. In comparison, $U' = \{[X_1 \cdot z_1 \equiv_\emptyset X_2 \cdot c, [X_1 = c, X_2 = x]]\}$ is not complete. x appears there as solution for c but it does not appear as a solution for d, though both c and $-d$ occur in one and the same row of C.

Once having a complete system of solutions corresponding to a given net it is very easy to calculate invariants. We simply collect the weight terms in the components of the invariant and compute their still unknown multiplicities from a system of linear equations on the integer numbers. This system grows from the requirement, that the sum of multiplicities in every equivalence class of $C^T \cdot I$ must equal 0.

Example. Consider the above complete system of solutions U. The weight terms which occur in U are c corresponding to the matrix term z_1, that is to the first place, furthermore x corresponding to the matrix terms c and d, that is belonging to the second place and finally d belonging to the last place. The raw invariant is $(a_1 \cdot c, a_2 \cdot x, a_3 \cdot d)^T$, where the a_i are the multiplicities which are still unknown. The application of I to C^T results in $(-a_1 \cdot c + a_2 \cdot c, -a_2 \cdot d + a_3 \cdot d)^T$. The following equations can be derived: $a_1 = a_2$ and $a_2 = a_3$. A generator set for all integer solutions of this system of equations is $a_1 = a_2 = a_3 = 1$. Therefore every invariant of the above raw kind is generated by $(1 \cdot c, 1 \cdot x, 1 \cdot d)^T$. The remaining problem is to compute complete systems of solutions. A suitable algorithmic way for obtaining complete systems of MGS is to start with a single equation and a solution and then repeatedly to search for weight terms which violate the completeness conditions. These violations can be removed either by instantiating solutions which are already contained in the considered system of solutions or by adding new equations with their solutions to the system.

Example The incidence matrix of the net in Figure 2(b) is $\begin{pmatrix} -c & \\ z_1 & -z_2 \\ & z_3 \end{pmatrix}$. We compute a complete system of solutions corresponding to this net. For this purpose we start with the equation $X_1 \cdot c \equiv_\emptyset X_2 \cdot z_1$ obtained from the first column of the incidence matrix, and an element of MGS, $[X_1 = x, X_2 = c]$. The completeness condition is violated for the weight term c, since there is no equation for z_2 which appears in the same row as z_1. Therefore we have to add an equation covering z_2. The only suitable equation for this purpose is $X_1 \cdot z_2 \equiv_\emptyset X_2 \cdot z_3$. This equation has only a singleton MGS, $[X_1 = y, X_2 = y]$. But with this solution the completeness condition is still violated. This violation can be removed by instantiating y to c. The resulting system of solutions consisting of $[X_1 = x, X_2 = c]$ for the first equation and $[X_1 = c, X_2 = c]$ for the second equation is obviously complete. This complete system of solutions is related to an invariant, namely $(x, c, c)^T$.

When we remove violations of the completeness condition by adding new equations it may happen that new violations of the condition appear. Therefore the termination of the algorithm cannot be guaranteed. Up to now we have no criterion when the computation of a complete system of solutions can be stopped, we also do not know whether for a given algebraic net there is always a finite generator set for the place invariants. But of course the algorithmic idea is sufficient to *enumerate* the complete systems of solutions. Therefore we will be able to enumerate generator sets of invariants which is often sufficient for the reachability test to be performed with place invariants. With the algorithm which we proposed we do not obtain *every* complete system of solutions.

Example Consider the net in Figure 2(b) and the following complete system of solutions U consisting of

$$\begin{array}{r} X_1 \cdot c \equiv_E X_2 \cdot z_1 / [X_1 = x, X_2 = c] \\ X_1 \cdot z_2 \equiv_E X_2 \cdot z_3 / [X_1 = c, X_2 = c] \\ X_1 \cdot c \equiv_E X_2 \cdot z_1 / [X_1 = y, X_2 = y] \\ X_1 \cdot z_2 \equiv_E X_2 \cdot z_3 / [X_1 = y, X_2 = y] \end{array}$$

This system is related to the invariant $(1 \cdot x + 1 \cdot y, 1 \cdot c + 1 \cdot y, 1 \cdot c + 1 \cdot y)^T$.

Obviously the proposed algorithm started on the first element of U would stop after having added the second element since no completeness restrictions are violated (this is just the computation discussed in the previous example). Starting the algorithm with another element always leads to complete systems of solutions which either contain only the first two or the last two elements of U. The reason is that the weight terms x and c on the one hand and y on the other are completely independent in the sense that equivalence classes in $C^T \cdot I$ containing terms which are made from applications of x and c to matrix terms are disjoint to the classes of terms which result from the application of a weight term y. Thus the invariant can be represented as $(1 \cdot x, 1 \cdot c, 1 \cdot c)^T + (1 \cdot y, 1 \cdot y, 1 \cdot y)^T$ where both **P**–indexed vectors are invariants.

Definition 5.2 *A complete system of solutions is called* **basic** *iff none of its proper subsets is a complete system of solutions.*

There are two important properties holding for basic complete systems of solutions: (1) Every invariant can be represented as a linear combination of invariants constructible from basic complete systems of solutions; (2) With the above algorithmic approach it is possible to find every basic complete system of solutions. We do not prove these claims here since a formal proof would require a couple of additional notations. However there are dual theorems for the case of transition invariants which have been proved in [Sc94]. The arguments used there can be established for place invariants as well. The only problem we still have to overcome is to decide if and how we can instantiate or concatenate solutions in a most general way such that two distinguished weight terms appearing in a solution are equivalent (that is, a violation of the completeness condition can be removed). The only way to force this equivalence is to instantiate the solutions or to concatenate them with other weight functions. As well as for the term equation problem we have a solution for this problem only for the case that the set of equations occurring in the specification is empty and the specified operation symbols are constant symbol or unary operation symbols. Consider the two equations

$$X_1 \cdot T_1 = \quad \cdots \quad = X_m \cdot T_m \tag{1}$$
$$X'_1 \cdot T'_1 = \quad \cdots \quad = X'_n \cdot T'_n \tag{2}$$

Assume we have a generator set for a set of solutions of both equations. We aim in a generator set for *all* solutions which are generated by the given generator sets and which satisfy the additional requirement that $X_1 = X'_1$. Every solution of this problem must be obtainable as an instance of an element of the first generator set and an element of the second generator set. Let $[X_1 = W_1, \cdots X_m = W_m]$ and $[X'_1 = W'_1, \cdots, X'_n = W'_n]$ be elements of the generator sets of solutions for the two equations. We consider 3 cases for W_1.
Case 1: W_1 is a parameter variable y. In this case it cannot be unified with W'_1 if W'_1 contains the argument variable x. In all other cases, that is W'_1 contains no variable or a parameter variable, it is possible to instantiate the solution of equation (1) by $[y \rightarrow W'_1]$. This solution and the original solution of equation (2) satisfy the considered requirements. Furthermore it is obvious that every pair of solutions which a) are instances of the two given solutions and b) satisfy the requirement $X_1 = X'_1$ are instances of the constructed solution.
Case 2: W_1 is ground term. Then there is no common solution if W'_1 contains

the argument variable x. If W_1' is a parameter variable y then we can swap the two solutions and return to case 1. If W_1' is a ground term, too, W_1 and W_1' can only be unified when $W_1 \sqsubseteq W_1'$ or $W_1' \sqsubseteq W_1$, since the only way to modify the solutions is to concatenate the solutions with weight terms. If $W_1 \sqsubseteq W_1'$ then obviously we have to concatenate the solution of equation (1) with $W_1' \setminus W_1$, while in the case $W_1' \sqsubseteq W_1$ the two weight functions can be unified by concatenating the solution of equation (2) with $W_1 \setminus W_1'$.
Case 3: W_1 contains the argument variable x. Then it is impossible to build common instances of both solutions when W_1' does not contain the argument variable x. Therefore the requirement $X_1 = X_1'$ can be satisfied only if W_1' contains x, too. In this case the only way to unify W_1 and W_1' is to concatenate them with weight terms. Thus it holds $W_1 \sqsubseteq W_1'$ or $W_1' \sqsubseteq W_1$ and the requirement can be satisfied by concatenating the solution of equation (1) with $W_1' \setminus W_1$ or the solution of equation (2) with $W_1 \setminus W_1'$, resp.
Note, that we did not consider the case that W_1 is a term which consists of the parameter variable y and unary operation symbols. This is however not necessary, since neither the MGS of equations considered in the previous section nor the instantiation of solutions discussed here provide such terms. The case consideration leads immediately to the following

Theorem 9 *Let S_1 and S_2 be sets of solutions of the equations $X_1 \cdot T_1 = \cdots = X_m \cdot T_m$ and $X_1' \cdot T_1' = \cdots = X_n' \cdot T_n'$. Then every pair of solutions $[X_1 = W_1, \cdots, X_m = W_m]$ and $[X_1' = W_1', \cdots, X_n' = W_n']$ which can be generated from S_1 and S_2, respectively and for which it holds $W_1 = W_1'$ can be generated from pairs of solutions which can be constructed by solutions of S_1 and S_2 according to the above case consideration.*

With these considerations we have a complete calculus to enumerate place invariants for nets based on simple specifications. In the next section we will present an example where all the steps of calculation are demonstrated again for the philosopher's example.

6 Example

We consider the philosopher's example as described earlier, but without equations in the specification. The incidence matrix of the net in Figure 1 is

	t_1	t_2	t_3	t_4
p_1	$-z$			z
p_2	z	$-z$		
p_3		z	$-z$	
p_4			z	$-z$
p_5	$-L(z)$	$-R(z)$	$L(z)$	$R(z)$

. In order to distinguish the different occurrences of syntactically identical terms in this matrix we superscribe the row and the column to every term. For instance we write z^{32} for the term z appearing in the row p_3 and the column t_2 of the matrix.

According to the algorithmical ideas discussed in the previous sections we have to compute complete systems of MGS for systems of term equations. For this purpose let us choose a starting equation. For t_1 we have the following starting equations: $X_1 \cdot z^{11} \equiv_\emptyset X_2 \cdot z^{21}$, $X_1 \cdot z^{11} \equiv_\emptyset X_2 \cdot L(z)^{51}$, $X_1 \cdot z^{21} \equiv_\emptyset X_2 \cdot L(z)^{51}$ and $X_1 \cdot z^{11} \equiv_\emptyset X_2 \cdot z^{21} \equiv_\emptyset X_3 \cdot L(z)^{51}$. Every of these four equations

as well as the corresponding equations for the other columns of the matrix are possible starting equations. If we aim in a complete generator set we have to perform the following steps beginning with every of these starting equations. We choose the third equation. The next step is to compute the system of MGS for this equation. It consists of the two solutions $[X_1 = L(x), X_2 = x]$ and $[X_1 = y, X_2 = y]$. The first solution is more interesting (of course, for a complete generator set we have to consider both solutions). We get the first system of solutions U containing the equation $X_1 \cdot z^{21} \equiv_\emptyset X_2 \cdot L(z)^{51}$ with its solution $[X_1 = L(x), X_2 = x]$. The completeness condition is violated twice, since first $L(x)$ is a solution corresponding to z^{21} but there is no equation for z^{22} appearing in the same row and second x is a solution corresponding to $L(z)^{51}$ and there is no equation for $R(z)^{52}$, $L(z)^{53}$ and $R(z)^{54}$. Let us try to remove the first violation. Therefore we need an equation covering z^{22}. This equation cannot involve $R(z)^{52}$ since the MGS of such an equation would provide for z^{22} either the weight term y or the weight term $R(x)$ both not having common instances with the currently considered weight term $L(x)$. The only possible equation to solve the problem is $X_1 \cdot z^{22} \equiv_\emptyset X_2 \cdot z^{32}$. The MGS of this equation consists of $[X_1 = x, X_2 = x]$ and $[X_1 = y, X_2 = y]$ the first of which has common instances with the solution already contained in U. Thereby the only way to obtain a most general common instance between the two solutions is to concatenate the second solutions with $L(x)$. Therefore the system of MGS currently is

$$\begin{array}{l} X_1 \cdot z^{21} \equiv_\emptyset X_2 \cdot L(z)^{51} / [X_1 = L(x), X_2 = x] \\ X_1 \cdot z^{22} \equiv_\emptyset X_2 \cdot z^{32} / [X_1 = L(x), X_2 = L(x)] \end{array}$$

The considered violation has been removed but a new violation of the completeness condition appeared for z^{32}. In the same manner we remove all the violations step by step including those which are caused by the inserted elements.

Choosing one of the different possibilities at every stage of the computation we result in the following value for U:

$$\begin{array}{l} X_1 \cdot z^{21} \equiv_\emptyset X_2 \cdot L(z)^{51} / [X_1 = L(x), X_2 = x] \text{ (starting eq.)} \\ X_1 \cdot z^{22} \equiv_\emptyset X_2 \cdot z^{32} / [X_1 = L(x), X_2 = L(x)] \text{ (viol. for } p_2) \\ X_1 \cdot z^{33} \equiv_\emptyset X_2 \cdot L(z)^{53} / [X_1 = L(x), X_2 = x] \text{ (viol. for } p_3) \\ X_1 \cdot z^{32} \equiv_\emptyset X_2 \cdot R(z)^{52} / [X_1 = R(x), X_2 = x] \text{ (viol. for } p_5) \\ X_1 \cdot z^{33} \equiv_\emptyset X_2 \cdot z^{43} / [X_1 = R(x), X_2 = R(x)] \text{ (viol. for } p_3) \\ X_1 \cdot z^{44} \equiv_\emptyset X_2 \cdot R(z)^{54} / [X_1 = R(x), X_2 = x] \text{ (viol. for } p_4 \text{ and } p_5) \end{array}$$

This system is complete; all weight terms which appear in U appear for every term of the corresponding row of the incidence matrix. Therefore we may start to construct an invariant. For this purpose we collect the weight terms from U and arrange them with unknown $\left(\begin{array}{c|c} p_2 & a_1 \cdot L(x) \\ p_3 & a_2 \cdot L(x) + a_3 \cdot R(x) \\ p_4 & a_4 \cdot R(x) \\ p_5 & a_5 \cdot x \end{array}\right)$. In order to determine the multiplicities $a_1, \cdots, a_5$, we remember that every equation in U corresponds to an equivalence class of the vector $C^T \cdot I$ for the invariant which we want to construct. The sum of multiplicities in every equivalence class must be

0. Therefore every equation in U provides an equation for the a_i. For instance, the first equation in U provides $1 \cdot a_1 + (-1) \cdot a_5 = 0$, since the application of $a_1 \cdot L(x)$ to $1 \cdot z^{21}$ and the application of $a_5 \cdot x$ to $(-1) \cdot L(z)^{51}$ form the equivalence class $[L(z)]$ in the component t_1 of $C^T \cdot I$ and the sum of multiplicities is just $a_1 - a_5$. Additionally we obtain $a_2 - a_1 = 0$, $-a_2 + a_5 = 0$, $a_3 - a_5 = 0$, $-a_3 + a_4 = 0$ and $-a_4 + a_5 = 0$. Every solution of this system of equations is a linear combination of $a_1 = \cdots = a_5 = 1$. Therefore every invariant with of computed structure can be generated from $\left(\begin{array}{c|c} p_2 & 1 \cdot L(x) \\ p_3 & 1 \cdot L(x) + 1 \cdot R(x) \\ p_4 & 1 \cdot R(x) \\ p_5 & 1 \cdot x \end{array}\right)$.

With this invariant one branch of the computation has been finished. The consideration of the other branches leads to the invariants

$$\left(\begin{array}{c|c} p_1 & 1 \cdot x \\ p_2 & 1 \cdot x \\ p_3 & 1 \cdot x \\ p_4 & 1 \cdot x \end{array}\right), \left(\begin{array}{c|c} p_1 & 1 \cdot y \\ p_2 & 1 \cdot y \\ p_3 & 1 \cdot y \\ p_4 & 1 \cdot y \end{array}\right), \left(\begin{array}{c|c} p_2 & 1 \cdot y \\ p_3 & 2 \cdot y \\ p_4 & 1 \cdot y \\ p_5 & 1 \cdot y \end{array}\right)$$

and some linear combinations of these invariants such as

$$\left(\begin{array}{c|c} p_1 & -1 \cdot R(x) \\ p_2 & 1 \cdot L(x) - 1 \cdot R(x) \\ p_3 & 1 \cdot L(x) \\ p_5 & 1 \cdot x \end{array}\right) = \left(\begin{array}{c|c} p_2 & L(x) \\ p_3 & L(x) + R(x) \\ p_4 & R(x) \\ p_5 & x \end{array}\right) - R(x) \circ \left(\begin{array}{c|c} p_1 & x \\ p_2 & x \\ p_3 & x \\ p_4 & x \end{array}\right).$$

Consequently our generator set is not minimal. Especially our algorithm cannot detect whether an invariant can be combined from other ones when the equivalence classes caused by the combined invariants are merged. For instance the application of both invariants above leads to the equivalence class $R(z)$ in the component t_3 of the resulting **T**–indexed vectors. But the weight terms which are responsible for this equivalence class cancel out each other in the combined invariant. Nevertheless the algorithm computes a small and fortunately a finite generator set of all invariants for the philosopher's net.

7 Conclusions

The problem of computing place invariants for algebraic nets can be traced back to equations between terms. We can solve these term equations in the case of equation free specifications with at most unary operation symbols. Though these restrictions are very hard several important problems can be specified this way. The generator sets we obtain with the proposed method are small in the following sense. A) We do not compute invariants which can be generated from other ones by simple concatenation or instantiation, we restrict ourselves to most general solutions and most general common instances. B) We avoid the computation of linear combinations, when the equivalence classes of the involved invariants do not influence each other. C) We only compute generator sets for the multiplicities of the weight terms occurring in a constructed invariant. Therefore, though the computed generator sets are not minimal, the computed set is suitable for the non–reachability test we aim in. Nevertheless a lot of problems remain. We only enumerate complete sets of solutions. We do

not have a termination criterion for the generation of such sets (for T–invariants we know that there is no such criterion). It requires a lot of additional work to be able to cover specifications with equations or arbitrary operation symbols. Perhaps it will be necessary to modify the concept of place invariants for this purpose. All in all the algorithm may be a suitable element of an analysis tool for algebraic nets and a useful supply for their human driven analysis.

References

[Co90] J.M. Couvreur. The general computations of flows for coloured nets. *Proc. of the 11th Int. Workshop on Appl. and Theory of Petri Nets*, 1990.

[CM91] J.M. Couvreur, J. Martinez. Linear invariants in commutative high level nets. *High-level Petri nets, Theory and Application*, 1991.

[EM85] H. Ehrig, B. Mahr. *Fundamentals of Algebraic Specifications 1. EATCS Monographs on Theor. Comp. Science 6.* Springer 1985.

[GL83] H. Genrich, K. Lautenbach. S–invariance in Pr/T–nets. *Informatik–Fachberichte*, (66), 1983.

[HC88] S. Haddad, J.M. Couvreur. Towards a general and powerful computation of flows for parameterized coloured nets. *Proc. of the 9th European Workshop on Appl. and Theory of Petri Nets*, 1988.

[Je81a] K. Jensen. Coloured Petri nets and the invariant–method. *Theor. Comp. Science*, 14:317–336, 1981.

[Je81b] K. Jensen. How to find invariants for coloured Petri nets. *LNCS*, 118:327–338, 1981.

[LP85] K. Lautenbach, A. Pagnoni. Invariance and duality in predicate/transition nets and in coloured nets. *Arbeitspapiere der GMD*, 132, 1985.

[MV87] G. Memmi, J. Vautherin. Analysing nets by the invariant method. In: Rozenberg, Brauer, Reisig (eds.), *Advances in Petri nets 1986*, pages 300–336, 1987.

[Re91] W. Reisig. Petri nets and algebraic specifications. *Theor. Comp. Science*, 80 (1991):1–34.

[Sc94] K. Schmidt. T–invariants of algebraic petri nets. *Informatik–Bericht der HU Berlin*, 31, 1994.

[S+91] M. Silva, J. Martinez, P. Ladet, H. Alla. Generalized inverses and the calculations of symbolic invariants for coloured nets. *High-level Petri nets, Theory and Application*, 1991.

[Va86] J. Vautherin. Parallel systems specification with colored Petri nets and abstract data types. *Proc. of the 7th European Workshop on ATPN* Oxford 1986, 5–23.

[Va87] J. Vautherin. Calculation of semi–flows for Pr/T–systems. In T. Murata, editor, *Int. Workshop on Petri Nets and Performance Models.* IEEE Computer Society Press, 1987.