

1989

An Algorithm for Computing S-Invariants for High Level Petri Nets

Chuang Lin

Dan C. Marinescu

Report Number:
89-860

Lin, Chuang and Marinescu, Dan C., "An Algorithm for Computing S-Invariants for High Level Petri Nets" (1989). *Department of Computer Science Technical Reports*. Paper 732.
<https://docs.lib.purdue.edu/cstech/732>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

AN ALGORITHM FOR COMPUTING S-INVARIANTS
FOR HIGH LEVEL PETRI NETS

Chuang Lin
Dan C. Marinescu

CSD-TR-860
February 1989

**AN ALGORITHM FOR COMPUTING S-INVARIANTS
FOR HIGH LEVEL PETRI NETS**

Chuang Lin

Institute of Information Science and Its Applications
State Economic Information Center
58 Sanlihe Road, Beijing
CHINA

Dan Cristian Marinescu

Computer Sciences Department
Purdue University
West Lafayette, IN 47907
USA

Abstract

Net invariants and reachability trees are used to investigate dynamic properties of Petri Nets. Both concepts have been generalized for different classes of High Level Petri Nets. In this paper we introduce the compound token and the token flow path concepts. Then we present an algorithm to compute the S -invariants of a High Level Petri Net using the compound token and the token flow path ideas and show that all S -invariants of an HLPN can be generated by a system of integer linear equations without unfolding the net.

Keywords: High-Level Petri Nets, algorithm, compound tokens, token flows paths, S -invariants.

1. INTRODUCTION

Petri Nets, PNs are one of the most interesting means for specification, modeling and analysis of concurrent systems. There are several classes of methods for analyzing the dynamic behavior of Petri Net models: methods based upon the investigation of the reachability set of the net, methods based upon homomorphic transformation of nets, and methods using linear net invariants.

The first class of methods depend upon the ability to construct the reachability set of a PN. For complex systems the reachability set of the Petri Net model is too large to allow any significant analysis because the computations involved become prohibitively expensive. Structural analysis based upon net invariants is an attractive alternative for complex Petri Net models. In this case the analysis is performed on local subnets while ignoring how the entire net behaves. The S -invariants are important for validation of properties as boundedness, mutual exclusions between place markings, liveness, etc.

High Level Petri Nets, HLPNs are a family of nets with individual tokens and token variables. Predicate/Transition Nets, Coloured Petri Nets and Relation Nets belong to this category of nets. The main advantage in using HLPNs is derived from their power of representation. Rather complex systems can be modeled as HLPNs in a succinct and readable way. While the descriptive power of High Level Petri Nets is unquestionable, the analysis of such nets raises difficult problems. General and efficient algorithms to construct the reachability trees and the invariants of High Level Petri Nets are unavailable.

The subject of invariants for Petri Nets is fairly well understood, see for example [4]. Genrich and Lautenbach [1] have generalized the concept of place invariants also called S -invariants, and transition invariants to different families of High Level nets. The problem of constructing S -invariants for High Level Petri Nets is more complex and no simple, general, and efficient algorithms are known.

Some concepts used successfully to obtain invariants for High Level Petri Nets are reviewed in the following. The concepts of quasi-invariants and proper-invariants are introduced in [2] for Predicate Transition Nets. The quasi-invariants can be systematically computed by the Gaussian Elimination algorithm [3], and contain free variables which have to be projected to obtain proper-invariants which give invariant assertion over markings. No general and simple method to find correct ways for projecting is known.

The concept of a weight-function is used to compute S -invariants for Coloured Petri Nets [6] through a sequence of transformation rules. But, in general, it is not possible to find all invariants using a simple algorithm.

The calculation of semi-flows for predicate transition systems [7], [8], [9] seems more practical, since leads to S -invariants which can be easily interpreted and can be obtained from a finite number of integer vectors. But this method involves additional overhead since all places in a net must have the same arity, and it is not easy to compute semi-flows of tokens with n -elements.

In this paper we introduce the concept of a **token flow path** in order to

- a. construct invariants with a straightforward interpretation, and
- b. bind free variables in a simple manner. In case of unary token flow paths transitions connect only two places.

The algorithm presented in this paper determines the S -invariants in an unary token flow path. The choice of the unary token flow path rather than the n -ary token flow path is well motivated. First of all, n -ary token flow paths with $n > 1$, seldom exist. Moreover, even when such paths exist, they can be easily constructed from the unary token flow paths.

The remainder of the paper is organized as follows. Section 2 reviews the definition of the High Level Petri Nets and introduces the compound token concept. In Section 3 the concept of token flow path is defined and an algorithm to find S -invariants in an unary token flow path is presented. An example is presented in Section 4.

2. DEFINITION AND NOTATIONS

This section presents definitions, terminology and notations which will be needed throughout the paper.

Denote by N the set of all non-negative integers and Z the set of all non-negative integers including zero.

Definition 2.1:

A **multi-set** P' is a function defined on a non-empty set P , $P' \in [P \rightarrow N]$.

Intuitively, a multi-set is a set which can contain multiple occurrences of the same element.

It should be pointed out that there are differences among definition of High Level Petri Nets given by the different authors. Here, we present a definition based upon the ones given in [6] and [2].

Definition 2.2:

A High-level Petri Net is an 8-tuple $H = (S, T, A, V, D, X, W, M_0)$, where

S : is a finite, non-empty set of **places**,

T : is a finite, non-empty set of **transitions**,

A : is a finite, non-empty set of **atomic colours**,

V : is a finite set of **variables**,

D : is a function defined on V such that for each variable $v \in V$, $D(v)$ is a set of atomic colours called the **domain** of the variable v .

X: is a colour function defined on $S \cup T$. $X(S)$ represents the set of place colours for a given maximal arity of places, n , $X(S) = \bigcup_{0 \leq k \leq n} A^k$ with $A^k = \{ \langle \rangle \}$. X associates with each place a set of possible token colours, i.e., $\forall p \in S, X(p) \subseteq X(S)$. $X(T)$ represents the set of transition colours. X attaches to each transition a set of possible occurrence colours, i.e., $\forall t \in T, X(t)$ is the set of substitutions of all variables appearing free in the immediate surrounding arc-expressions of t and $X(T) = \bigcup_{t \in T} X(t)$.

W: is an arc function defined on $(S \times T) \cup (T \times S)$. It indexes family of multi-sets over $\bigcup_{0 \leq k \leq n} (A \cup V)^k$, i.e.,

$$\forall (x,y) \in (S \times T) \cup (T \times S), \quad W_{x,y}: \bigcup_{0 \leq k \leq n} (A \cup V)^k \rightarrow N.$$

M_0 : is called the initial marking of H .

It is known that each High Level Petri Net with a finite set of colours can be transformed into an equivalent Place/Transition Net obtained through unfolding each place p into the set of places $\{(p,a) \mid a \in X(p)\}$, and unfolding each transition t into the set of transitions $\{(t,\sigma) \mid \sigma \in X(t)\}$. Sometime instead of a transition t reference to a step $(t,\sigma) \mid \sigma \in X(t)$ is made. A step is regarded as a generic transition.

Definition 2.3

The incidence matrix of a High Level Petri Net H , is the matrix $C = (C_{p,t})$ for all $p \in S$, and $t \in T$ with $C_{p,t}$ defined as $C_{p,t} = W_{t,p} - W_{p,t}$.

Thus

$$C_{p,t}: \bigcup_{0 \leq k \leq n} (A \cup X)^k \rightarrow Z.$$

Let $W_{p,t}(\sigma)$ be the multi-set obtained from $W_{p,t}$ by substituting the free variables by atomic colours according to σ . For instance, if

$$\sigma = (x \leftarrow b, y \leftarrow a) \text{ and } W_{p,t} = \langle a,x \rangle + \langle x,y \rangle \text{ then } W_{p,t}(\sigma) = \langle a,b \rangle + \langle b,a \rangle.$$

Definition 2.4

The transition t is enabled at the marking M iff

$$\exists \sigma \in X(t) \text{ such that } W_{p,t}(\sigma) \leq M(p) \quad \forall p \in S .$$

We will say that the step (t,σ) , rather than transition t is enabled if a colour function σ exists, i.e. $\exists \sigma \in X(t)$. The step (t,σ) is not enabled if a colour function σ does not exist, i.e. $(t,\sigma) = 0$ if $\sigma \notin X(t)$.

Definition 2.5

When a step (t, σ) is enabled at M_1 , it can fire and transform marking M_1 into a **directly reachable marking** M_2 defined as

$$M_2(p) = M_1(p) - W_{p,t}(\sigma) + W_{t,p}(\sigma) \quad \forall p \in S.$$

Denote the t -th column of the incidence matrix C by C^t and note that C^t is associated with transition t . To underline the fact that M_2 is reached from M_1 when transition t fires, the definition of the follower marking presented above can be rewritten as

$$M_2 = M_1 + C^t(\sigma).$$

Let q be a **step sequence**, the equivalent of a firing sequence

$$q = \{(t, \sigma_1), (t_2, \sigma_2), \dots, (t_k, \sigma_k)\}.$$

Definition 2.6

A marking M is **reachable** from the initial marking M_0 iff a step sequence exists such that

$$M = M_0 + C * q.$$

Here $C * q$ is defined as

$$C * q = \sum_{(t, \sigma) \in q} C * (t, \sigma) = \sum_{(t, \sigma) \in q} C^t(\sigma).$$

Definition 2.7

The **reachability set** corresponding to the initial marking M_0 is denoted as $[M_0 >$ and it is defined as the set of all markings which are reachable from some initial marking M_0 .

Definition 2.8:

\mathcal{L} is a linear function on the reachability set. $\mathcal{L}: [S \rightarrow (X(S) \rightarrow N)] \rightarrow Z$; \mathcal{L} is an **S-invariant** if $\mathcal{L}(M) = \mathcal{L}(M_0)$ for all $M \in [M_0 >$.

In a High Level Petri Net the tokens flowing through the system are distinguishable from one another. Such a token has an list of attributes associated with it and it will be called a **compound token**. A compound token can be regarded as a collection of unary tokens, tokens with one attribute only.

In the arc labeling a compound token a is described by the n -tuple $\langle a_1, \dots, a_n \rangle$, with a_i , $1 < i < n$, an atomic colour (or variable). a_i is the projection of the compound

token a along the i -th dimension of the colour space. To have a unitary representation we consider an unary token as an n -tuple with one element only, along one of the directions of the colour space.

The $*$ notation is used to indicate colour dimension(s) which are not relevant for the current flow of the compound token. For example, the n -tuples $\langle a, b, c \rangle$ and $\langle b \rangle$ are represented by the notation $(b,*)$ whenever only the colour b is of interest. The notation $M(p)(a,k)$ indicates that place p contains tokens with the atomic colour of interest in the k -th position.

3. TOKEN FLOW PATHS AND S-INVARIANTS FOR HIGH LEVEL NETS

To compute S -invariants for a High Level Petri Net we introduce the token flow path concept and describe an algorithm to construct the S -invariants. The introduction of the token flow path concept simplifies the computation of S -invariants for High Level Petri Nets and gives them a clear interpretation. Finally we give an example to illustrate the algorithm.

The Token Flow Path

Let $H = (S, T, A, V, D, X, W, M_0)$ be a High Level Petri Net. Then the associated Petri Net $|H|$ is defined as:

$|H| = (S, T, |W|, |M_0|)$ with:

$$|W|(x,y) = \begin{cases} 1 & |W(x,y)| > 1 \\ 0 & |W(x,y)| = 0 \end{cases} \quad (x,y) \in (S \times T) \cup (T \times S) \quad (3.1)$$

$$|M_0|(p) = |M_0(p)| \quad \forall p \in S \quad (3.2)$$

Informally $|H|$ is obtained from H by omitting the colours of tokens and the number of tokens.

The incidence matrix of $|H|$ is

$$|C| = (|c|_{p,t})$$

for all $p \in S, t \in T$ with

$$|c|_{p,t} = |W|_{t,p} - |W|_{p,t}.$$

$|H|$ is an ordinary Petri Net and its S -invariants can be computed using for example the Martinez-Silva [3] algorithm. After computing the S -invariants of $|H|$ we eliminate all non-minimal support S -invariants and focus our attention upon minimal support S -invariants. Let us call S_f the set of places in such an S -invariant. The places in S_f are connected through a set of transitions and arcs and they form a subnet f of H . Such a subnet f is called a **token flow path**, $f = (S_f, T_f, A_f, D, X_f, W_f, M_0^f)$. An important property of the token flow path is its closure. This means that transition $t, \forall t \in T_f$

connects places p_i , and p_j in S_f , $p_i, p_j \in S_f$.

First we compute the S -invariants of $|H|$ and note that the corresponding elements in the S -invariant of $|H|$ are strictly positive.

Algorithm 1 (Martinez-Silva):

1. $A := |C|$; $D := I_n$ (n is place id).
2. Repeat for $i = 1$ until $i = m$ (m is transition id).
 - 2.1 Append to the matrix $[D | A]$ every rows resulting as a non-negative line combination of row pairs from $[D | A]$ that annul the i -th column of A .
 - 2.2 Eliminate from $[D | A]$ the rows in which the i -th column of A is non-null.

It is guaranteed that this method produces all minimal support invariants of $|H|$ according to the following theorem

Theorem 3.1 (Martinez-Silva) Algorithm 1 generates all the minimal support invariants of $|H|$ and each invariant is obtained from a subnet $|f|$.

The algorithm to compute the S -invariants of a High Level Petri Net H are based upon the following theorem

Theorem 3.2 Let I be an S -invariant of H . Then I can be expressed as a linear combination of S -invariants of some token flow paths of H .

Proof: Unfolding H and all token flow paths we obtain an equivalent net H' and equivalent subnets. I is equivalent with I' an S -invariant of H' . I' can be expressed as a linear combination of invariants from some equivalent subnets of some token flow paths because all subnets form a basis of S -invariants of H' according to the Theorem 3.1.

Computation of S-Invariants

Call E_f the subset of atomic colours present in the token flow path f , i.e., $a \in E_f \iff \exists p \in S_f \wedge c_{p,i}((a, *)) \neq 0$. The complement of E_f is denoted by NE_f . Call NE_f^+ the set containing all colours from NE_f and the additional colours from E_f .

Theorem 3.3 (Genrich): Let C be a HLPN-matrix and let p be a place whose arity is $m > 1$. For some k with $1 < k < m$, let $C' = |C|_k^p$ designate the result of projecting all tokens of row C_p , in C along the k -th position. Let $I': S \rightarrow [X(S) \rightarrow N]$ be a variable-free solution of $I' * C' = 0$. Then for every monomial $v: X(S) \rightarrow N$, the linear function I'_v defined by $I'_v(M) = (I' * |M|_k^p) \neq e^v$ is an S -invariant.

Lemma 3.4 (Genrich): The total projection of C , $|C|$ is the incidence matrix of an ordinary Place/Transition Net that represents the mere quantitative aspect of HLPN. The

total projection $|H|$ of every solution of $I * C = 0$ is an S -invariant of the P/T Net.

The following propositions can be proved using Theorem 3.3 and Lemma 3.4. The S -invariants of the token flow path f will be computed by the following formulas.

Proposition 1: Let $u = ((X_p), (Y_{p,a,k}) \mid p \in S_f, a \in E_f, k \in (K))$ be a solution of

$$\begin{cases} \forall a \in E_f \forall t \in T_f \sum_{p \in S_f} (X_p \mid c_{p,t} \mid + \sum_{k \in K} Y_{p,a,k} c_{p,t}((a, *))) = 0 \\ \forall x \in X \ a \in D(x) \sum_{p \in S_f} Y_{p,a,k} c_{p,t}((x, *)) = 0 \end{cases} \quad (P1)$$

The corresponding invariant is

$$\forall M \in [M_0 > \sum_{p \in S_f} (X_p \mid M(p) \mid + \sum_{k \in K} Y_{p,a,k} M(p)(a, K)) = c^{t,e} \text{ (constant)}$$

There may be two kinds of solutions in the previous expression, one the S -invariants of the High Level Petri Net and the other the S -invariants of the P/T Net.

Proposition 2: Let $w = ((Z_{p,a,k} \mid p \in S_f, a \in NE_f^+, k \in K))$ be a solution of

$$\begin{cases} \forall a \in NE_f^+ \ \forall t \in T \ \forall x \in X_f \ a \in D(x) \\ \sum_{p \in S_f} Z_{p,a,k} c_{p,t}((x, *)) = 0 \end{cases} \quad (P2)$$

The corresponding invariant is

$$\forall M \in [M_0 > \sum_{p \in S_f} \sum_{k \in K} Z_{p,a,k} M(p)(a, k) = c^{t,e} \text{ (constant)}$$

All S -invariants of a HLPN can be constructed from the S -invariants of every token flow path which form a basis of the solution.

An Example

Figure 1a, presents a HLPN H . We have $A = (a, b, c)$, $V = (x, y, z)$ and $D(x) = \{a, b\}$, $D(y) = \{b, c\}$, $D(z) = \{a, c\}$. The incidence matrix of H , denoted by C is presented in Figure 1b.

We apply Algorithm 1 to $|H|$, and obtain two token flow paths f and g , shown in Figure 2a and Figure 2b. Applying the formula in Proposition 1 to the flow path f we obtain the equation groups as follows:

$$\begin{cases} X_{p_1}^{*(-4)} + Y_{p_1, c, 1}^{*(-3)} + X_{p_3}^{*4} + Y_{p_3, c, 3}^{*1} = 0 \\ Y_{p_1, c, 1}^{*(-3)} + X_{p_4}^{*1} = 0 \\ Y_{p_3, c, 3}^{*(-3)} + Y_{p_4, c, 3}^{*1} = 0 \end{cases} \quad (1)$$

$$\begin{cases} X_{p_1}^{*(-4)} + X_{p_3}^{*4} + Y_{p_3,b,2}^{*1} = 0 \\ Y_{p_3,b,2}^{*3} = 0 \\ X_{p_3}^{*(-3)} + X_{p_4}^{*1} = 0 \end{cases} \quad (2)$$

$$\begin{cases} X_{p_1}^{*(-4)} + X_{p_3}^{*4} + Y_{p_{3,a,1}}^{*1} = 0 \\ Y_{p_{3,a,1}}^{*3} = 0 \\ X_{p_2}^{*(-3)} + X_{p_4}^{*1} = 0 \end{cases} \quad (3)$$

Then, we can obtain two S -invariants including one special one:

$$|M(p_1)| + 3|M(p_3)| + 9|M(p_4)| + 3M(p_1)(c, 1) + M(p_3)(c, 3) + 3M(p_4)(c, 3) = C^{te}$$

$$|M(p_1)| + |M(p_3)| + 3|M(p_4)| = C^{te}$$

From Proposition 2, we deduce the following equation group the flow path f :

$$\begin{cases} Z_{p_1,a,1}^{*(-1)} + Z_{p_3,a,3}^{*3} = 0 \\ Z_{p_3,a,3}^{*(-3)} + Z_{p_4,a,3}^{*1} = 0 \end{cases}$$

and the corresponding invariant:

$$3M(p_1)(a, 1) + M(p_3)(a, 3) + 3M(p_4)(a, 3) = C^{te}$$

In the same way, we can deduce the following equation groups and S -invariants for the flow path g :

$$\begin{cases} X_{p_2}^{*(-3)} + X_{p_3}^{*4} + Y_{p_3,a,1}^{*1} = 0 \\ Y_{p_3,a,1}^{*3} + Y_{p_2,a,1}^{*(-2)} = 0 \\ X_{p_3}^{*(-3)} + X_{p_4}^{*1} = 0 \\ Y_{p_3,a,1}^{*(-3)} + Y_{p_4,a,1}^{*1} = 0 \end{cases} \quad (1)$$

and the corresponding invariant:

$$2|M(p_2)| + |M(p_3)| + 3|M(p_4)| + 3M(p_2)(a, 1) + 2M(p_3)(a, 1) + 6M(p_4)(a, 1) = C^{te}$$

$$\begin{cases} X_{p_2}^{*(-3)} + X_{p_3}^{*4} + Y_{p_3,b,2}^{*1} = 0 \\ Y_{p_3,b,2}^{*3} + Y_{p_2,b,1}^{*(-1)} = 0 \\ Y_{p_2,b,1}^{*(-2)} + Y_{p_3,b,1}^{*3} = 0 \\ X_{p_3}^{*(-3)} + X_{p_4}^{*1} = 0 \\ Y_{p_3,b,1}^{*(-3)} + Y_{p_4,b,1}^{*1} = 0 \\ Y_{p_3,b,2}^{*(-3)} + Y_{p_4,b,2}^{*1} = 0 \end{cases} \quad (2)$$

and the corresponding invariant:

$$3|M(p_2)| + 2|M(p_3)| + 6|M(p_4)| + 3M(p_2)(b, 1) + 2M(p_3)(b, 1) + M(p_3)(b, 2) + 6M(p_4)(b, 1) + 3M(p_4)(b, 2) = C^{te}$$

$$\begin{cases} X_{p_2}^{*(c-3)} + X_{p_3}^{*4} + Y_{p_3,c,3}^{*1} = 0 \\ Y_{p_3,c,3}^{*3} = 0 \\ X_{p_3}^{*(c-3)} + X_{p_4}^{*1} = 0 \end{cases} \quad (3)$$

and the corresponding invariant:

$$4|M(p_2)| + 3|M(p_3)| + 9|M(p_4)| = C^{te}$$

$$\begin{cases} Z_{p_2,c,1}^{*(c-1)} + Z_{p_3,c,2}^{*3} = 0 \\ Z_{p_3,c,2}^{*(c-3)} + Z_{p_4,c,2}^{*1} = 0 \end{cases} \quad (4)$$

and the corresponding invariant:

$$3M(p_2)(c, 1) + M(p_3)(c, 2) + 3M(p_4)(c, 2) = C^{te}$$

4. CALCULATION OF S-INVARIANTS FOR THE PHILOSOPHER SYSTEM

To illustrate the simplicity and the power of the algorithm described in this paper we consider the philosopher system, consisting of five philosophers who alternatively think and eat. There are only five chopsticks on a circular table and there is one chopstick between any two philosophers. Each philosopher needs to use the two chopsticks adjacent to him when he eats. Obviously two neighbors cannot eat at the same time. The philosopher system can be described by the net shown in Figure 3. The model has fifteen places and ten transitions, all indexed on variable i , $i \in [1,5]$ in the following description:

- T_i Is the "thinking" place. If T_i holds a token, the i -th philosopher is thinking or waiting for chopsticks.
- E_i Is the "eating" place. If E_i holds a token, the i -th chopstick is free.
- F_i Is the "free chopsticks" place. If F_i holds a token, the i -th chopstick is free.
- G_i Is the "getting chopsticks" transition.
- R_i Is the "releasing chopsticks" transition.

For this Petri Net, we can use algorithm 1 to get the following ten S -invariants which are linearly independent and form a basis:

- (1) $M(T_1) + M(E_1) = 1$
- (2) $M(T_2) + M(E_2) = 1$
- (6) $M(E_1) + M(E_5) + M(F_1) = 1$
- (7) $M(E_1) + M(E_2) + M(F_2) = 1$

$$\begin{array}{ll}
 (3) & M(T_3) + M(E_3) = 1 \\
 (4) & M(T_4) + M(E_4) = 1 \\
 (5) & M(T_5) + M(E_5) = 1 \\
 (8) & M(E_2) + M(E_3) + M(F_3) = 1 \\
 (9) & M(E_3) + M(E_4) + M(F_4) = 1 \\
 (10) & M(E_4) + M(E_5) + M(F_5) = 1
 \end{array}$$

If we fold this PN, we get a model of the system described by the HLPN in Figure 4. In this model the place T stands for the set $\{T_i\}$ and F for $\{F_i\}$, the transition G stands for the set $\{G_i\}$, and the transition R represents the set $\{R_i\}$ with $i \in [1,5]$.

From the incidence matrix C of this HLPN, we obtain the token flow paths shown in Figure 5.

From Proposition 1 and Proposition 2, we compute the following ten S -invariants for the HLPN system without unfolding the net. They are equivalent with those of the P/T system.

$$\begin{array}{ll}
 (1) & M(T)(p_1,1) + M(E)(p_1,1) = 1 \\
 (2) & M(T)(p_2,1) + M(E)(p_2,1) = 1 \\
 (3) & M(T)(p_3,1) + M(E)(p_3,1) = 1 \\
 (4) & M(T)(p_4,1) + M(E)(p_4,1) = 1 \\
 (5) & M(T)(p_5,1) + M(E)(p_5,1) = 1 \\
 (6) & M(E)(f_1,2) + M(E)(f_1,3) + M(F)(f_1,1) = 1 \\
 (7) & M(E)(f_2,2) + M(E)(f_2,3) + M(F)(f_2,1) = 1 \\
 (8) & M(E)(f_3,2) + M(E)(f_3,3) + M(F)(f_3,1) = 1 \\
 (9) & M(E)(f_4,2) + M(E)(f_4,3) + M(F)(f_4,1) = 1 \\
 (10) & M(E)(f_5,2) + M(E)(f_5,3) + M(F)(f_5,1) = 1
 \end{array}$$

The ten S -invariants presented above can be re-written in a compact form as

$$M(T)(p_i,1) + M(E)(p_i,1) = 1 \quad (1)$$

$$M(E)(f_i,2) + M(E)(f_i,3) + M(F)(f_i,1) = 1 \quad (2)$$

for $i \in \sim[1,5]$

5. CONCLUSIONS

In this paper we have introduced the concept of a compound token and token flow path and based upon these two concepts and we have presented a simple and efficient algorithm for the computation of S -invariants of High-Level Petri Nets. Using our formalism the S -invariants have a simple interpretation.

A software package based upon this algorithm has been implemented to prove the viability and simplicity of the algorithm.

REFERENCES

- [1] H.J. Genrich, and K. Lautenbach, "System Modeling with High Level Petri Nets", Theoretical Computer Science, 13, pp. 109-136, 1981.
- [2] H.J. Genrich, "Predicate/Transition Nets", Lecture Notes in Computer Science, vol. 254, pp. 207-247, 1986.
- [3] R. Kujansuu and M. Lindvist, "Efficient Algorithms for Computing S -Invariants for Predicate/Transition Nets", Proceeding of the 5th European Workshop on

Applications and Theory of Petri Nets, Aarhus University, pp. 156-173, 1984.

- [4] K. Lautenbach, "Linear Algebraic Techniques for Place/Transition Nets", Lecture Notes in Computer Science, vol. 254, pp. 142-167, 1986.
- [5] J. Martinez and M. Silva, "A Simple and Fast Algorithm to Obtain all Invariants of Generalized Petri Nets", Informatik-Fachbrichte 52, (C. Girault and W. Reisig, eds.), Springer-Verlag, pp. 301-303, 1982.
- [6] K. Jensen, "Coloured Petri Nets", Lecture Notes in Computer Science, vol. 254, pp. 248-299, 1986.
- [7] J. Vautherlin, "Calculation of Semi-Flows for Pr/T-systems", Proceedings of International Workshop on Petri Nets and Performance Models, University of Wisconsin-Madison, pp. 174-183, 1987.
- [8] G. Memmi and J. Vautherlin, "Analyzing Nets by the Invariant Method", Lecture Notes in Computer Sciences, Vol. 254, pp. 300-336, 1986.
- [9] J. Vautherlin and G. Memmi, "Computation of Flows for Unary Predicates/Transitions Nets", Advances in Petri Nets 1984, LNCS 188, (G. Rosenberg, ed.), pp. 307-327, Springer-Verlag, 1985.

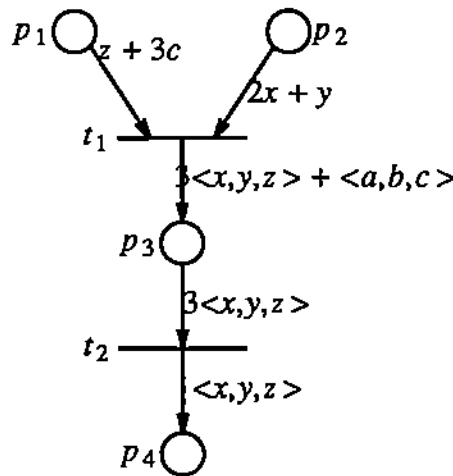


Figure 1a.

C	t_2	t_2
p_1	$-(z + 3c)$	
p_2	$-(2x + y)$	
p_3	$3\langle x, y, z \rangle + \langle z, b, c \rangle$	$-3\langle x, y, z \rangle$
p_4		$\langle x, y, z \rangle$

Figure 1b.

Figure 1. An example of a High Level Petri Net

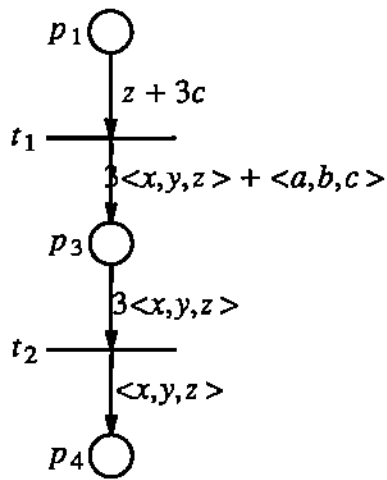


Figure 2a.

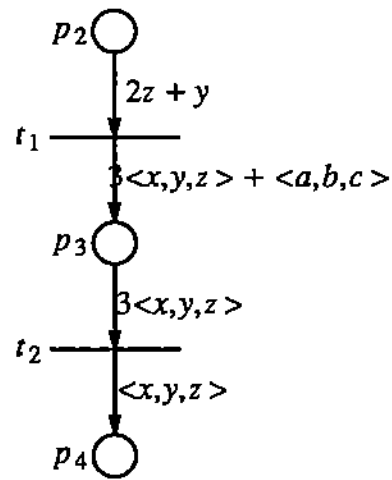


Figure 2b.

Figure 2. The token flow paths for the HLPN in Figure 1

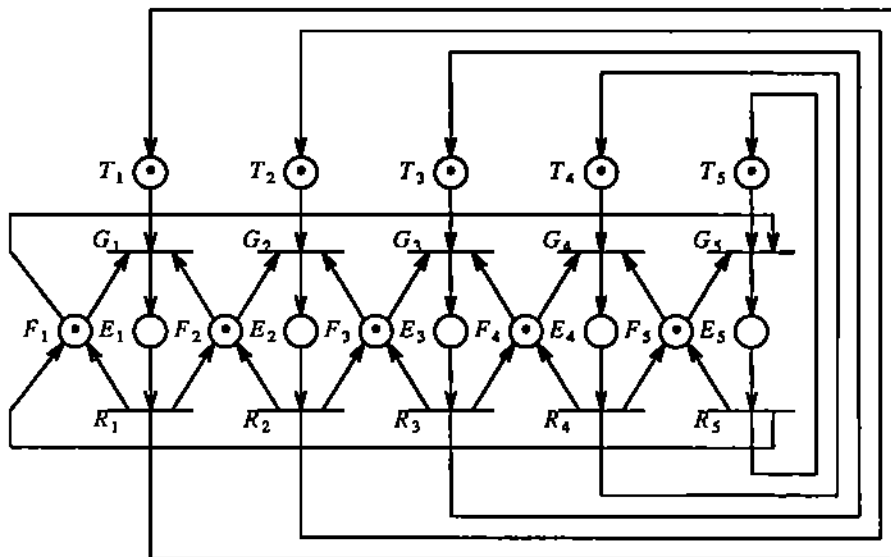
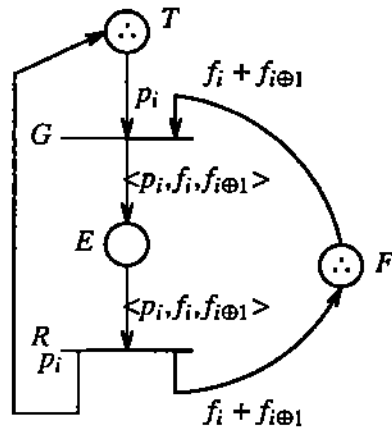


Figure 3. The Petri Net model of the philosopher system



C	G	R
T	$-p_i$	p_i
E	$\langle p_i, f_i, f_{i\ominus 1} \rangle$	$-\langle p_i, f_i, f_{i\ominus 1} \rangle$
F	$f_i + f_{i\ominus 1}$	$f_i + f_{i\ominus 1}$

Figure 4. The High Level Petri Net model of the philosopher system

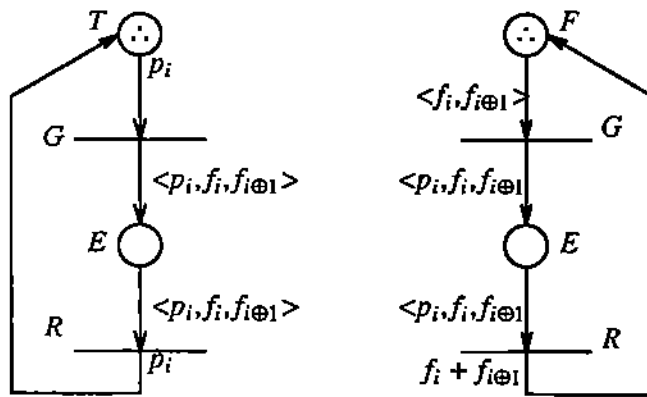


Figure 5a

Figure 5b

Figure 5. Two token flow paths of the philosopher system