ISSN 0105-8517

HOW TO FIND INVARIANTS FOR COLOURED PETRI NETS

by

Kurt Jensen

DAIMI PB-120
May 1980

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55

SUMMARY

This paper shows how invariants can be found for coloured Petri nets. We define a set of transformation rules, which can be used to transform the incidence-matrix, without changing the set of invariants.

1. INTRODUCTION

In [2] coloured Petri nets are defined as a generalisation of place/transition-nets, and it is shown how to generalise the invariant-concept, [3], to coloured Petri nets. The elements in the involved matrices are no longer integers but functions, and matrix-multiplication is generalised to involve composition/application of these functions. In [2] it is shown how to use invariants when proving various properties for the considered systems. In the present paper it will be shown how to find invariants by a sequence of transformations mapping the incidence-matrix into gradually simpler matrices with the same set of invariants. The present paper is a continuation of [2], and it will use the definitions and notations from [2] without further explanation.

In section 2 we define four transformation rules, which can be used to transform the incidence-matrix of a coloured Petri net. The four transformation rules are inspired by the method of Gauss-elimination, which is used for matrices, where all elements belong to a field. We prove that the transformation rules are sound, i.e. they do not change the set of invariants.

The matrix-elements for coloured Petri nets are not contained in a field, but only in a non-commutative ring, and thus division of two elements may be impossible. For this situation no general algorithm is known to solve homogeneous matrix-equations. Thus we cannot expect our set of transformation rules to be complete, i.e. it is in general not possible to find all invariants only by means of the rules.

Although our set of transformation rules is not complete, it often allows us to transform the incidence-matrix to such a degree, that a number of invariants immediately can be found by inspection of the simplified matrix. In sections 3-5 we describe three different systems by means of coloured Petri nets, and we use the transformation rules to find invariants. Section 6 is a conclusion and we summarise the results from our three examples.

When a coloured Petri net is used to describe a system the corresponding incidence-matrix normally has the following properties:

- it is a sparse matrix
- there is a high degree of dependency between the individual columns
- there are several solutions for the homogeneous matrix-equation
- many of the matrix-elements are simple commutative functions, e.g. identity-functions
- it is not a square-matrix.

Our transformation rules are designed to benefit from these properties, and they will not be adequate for other more general kinds of matrices.

2. TRANSFORMATION RULES

In this section we define the four transformation rules and prove their soundness.

In an incidence-matrix each row corresponds to a single place. We shall, however, define our transformation rules on a more general form of matrices, where each row may have a set of places attached. Each place is attached to at most one row, and it carries a weightfactor indicating how to translate solutions for the homogeneous matrix-equation into invariants (details will be defined later).

Let $\mathbb{D}$ be $\mathbb{N}$ or $\mathbb{Z}$. In [2] we considered functions of the form $f \in [A \to [B \to \mathbb{D}]]$ and the linear extension $f \in [[A \to \mathbb{D}] \to [B \to \mathbb{D}]]$ defined to satisfy $f(g)(b) = \sum_{a \in A} g(a)\ f(a)(b)$ for all $g \in [A \to \mathbb{D}]$ and $b \in B$. However, to guarantee convergence of the summation, it is necessary to replace any set of the form $[C \to \mathbb{D}]$ by its subset $[C \to \mathbb{D}]_f$ containing only those functions h, where the support $\{c \in C \mid h(c) \neq 0\}$ is finite. For finite C we have $[C \to \mathbb{D}]_f = [C \to \mathbb{D}]$.

Let P be the set of places of a coloured Petri net and for each $p \in P$ with colour-set $C(p)$ define $D(p) = [C(p) \to \mathbb{Z}]_f$. A matrix (with places and weightfactors attached) is wellformed (over P) iff it has the following properties:

a) All matrix-elements are linear functions.

b) Each column has attached a nonempty set C, and each element in the column has $[C \to \mathbb{Z}]_f$ as domain.

c) Each row has attached a nonempty set C, and each element in the row has $[C \to \mathbb{Z}]_f$ as range.

d) All places attached to rows are elements of P.

e) Each place is attached to at most one row.

f) Each place p, attached to a row with range D (see below), has a weightfactor, which is a linear function from D(p) into D.

The domain (range) of a column (row) in a wellformed matrix is defined as the domain (range) of its elements.

For each coloured Petri net the incidence-matrix is wellformed over the set of places in the net. In each step of our transformations we shall assume the current matrix to be wellformed, and it can be proved that our transformation rules preserve this property. But first we define how to translate the solutions of a homogeneous matrix-equation of a wellformed matrix over P, into invariants over P.

It should be remembered that we consider homogeneous matrix-equations of the form $u \circledast W = O$, where the unknown vector u has an element u(r) for each row r in the matrix W. In particular this means that our generalisation of Gauss-elemination operates on columns instead of rows.

Let u be a solution to the homogeneous matrix-equation of a wellformed matrix over a set of places P. The corresponding invariant v is defined by the following equation satisfied for each place $p \in P$:

$$v(p) = \begin{cases} u(r) \circ w & \text{if} \quad p \text{ is attached to row } r \text{ with weightfactor } w \\ O & \text{if} \quad p \text{ is not attached to any row} \end{cases}$$

An invariant v covers a place p if the weight v(p) differs from the zero-function O.

To define our transformation rules we need the following definition, which may be motivated by a careful inspection of the proof for our soundness-theorem. A function of the form $f \in [[A \to \mathbb{Z}]_f \to [B \to \mathbb{Z}]_f]$ is pseudosurjective iff $\forall b \in B \; \exists g \in [A \to \mathbb{Z}]_f \; \exists z \in \mathbb{Z} - \{0\} \; [f(g) = zb]$. Surjectivity implies pseudosurjectivity.

Before each transformation-step we assume the current matrix to be of the form $W = (W_{ij})_{1 \le i \le n,\, 1 \le j \le m}$ where $1 \le n, m < \infty$.

Transformation 1

If column j (i.e. the functions $(W_{ij})_{1 \le i \le n}$) has domain A_j, D is a non-empty set, $A_j' = [D \to \mathbb{Z}]_f$ and $h \in [A_j' \to A_j]$ is a pseudosurjective linear function, replace column j by $(W_{ij} \circ h)_{1 \le i \le n}$ and replace domain A_j by A_j'.

Transformation 2

If two different columns j and k have domains A_j and A_k and $h \in [A_j \to A_k]$ is a linear function, replace column j by $(W_{ij} + W_{ik} \circ h)_{1 \le i \le n}$.

Transformation 3

If all elements in column j are zero-functions, remove column j.

Transformation 4

If all elements in column j are zero-fucctions, except two different elements W_{ij} and W_{kj}, which satisfy $W_{ij} = h$ and $W_{kj} = -g \circ h$, where g and h are linear functions and h is pseudosurjective, replace row k by $(g \circ W_{ij} + W_{kj})_{1 \le j \le m}$. For each place p attached to row i with weightfactor w, give p weightfactor $w \circ g$, and if $w \circ g$ differs from the zero-function, attach p to row k. Remove row i and column j.

Transformations 1 and 2 are generalisations of the rules for Gauss-elimination. They can be used to simplify the matrix-elements, but do not change the size of the matrix. Columns and rows can be removed by transformation 3 and by transformation 4.

Column j is a linear combination of a set of columns A iff there exists a family of linear functions $\{g_a \mid a \in A\}$ such that $W_{ij} = \sum_{a \in A} W_{ia} \circ g_a$ for all $i \in 1..n$.

From transformations 2 and 3 we can derive a fifth transformation rule. It allows us to remove any column, which is a linear combination of a set of other columns. It should be noted, that a column j may be a linear combination of a column k, without k being a linear combination of j.

Transformation 4 may seem complicated, but in most applications g will be a very simple function.

If g is the identity-function we have a column with two non-zero elements, which satisfy $W_{ij} = -W_{kj}$. Then column j corresponds to an equation, where any solution u must satisfy u(i) = u(k). Transformation 4 allows us to add row i and row k. The set of places attached to the new row is the union of those attached to the old. All weightfactors are unaltered.

If g is the zero-function we have a column with only one non-zero element W_{ij}. Then column j corresponds to an equation, where any solution u must satisfy u(i) = 0. Transformation 4 allows us to remove row i, together with the places attached to it.

A matrix W' (with domains, ranges, places and weightfactors attached) is obtainable from an incidence-matrix W of a coloured Petri net iff there exists a sequence of transformations of types 1-4, which transforms W into W'.

The four transformation rules are independent, i.e. omission of any of them would decrease the set of obtainable matrices.

Proposition

Any matrix obtainable from an incidence-matrix with places P is wellformed over P.

Proof Check that each transformation rule preserves wellformedness.

□

Theorem (soundness)

Let W' be a matrix obtainable from the incidence-matrix W of a coloured Petri net. Then W' and W have exactly the same set of invariants.

Proof For each type of transformation rule we prove that a single application does not change the set of invariants. Then a simple induction-argument completes the proof.

Transformation 1

We shall prove $\sum_{i=1}^{n} v_i \circ W_{ij} = O_1 \Leftrightarrow \sum_{i=1}^{n} v_i \circ (W_{ij} \circ h) = O_2$ where O_1 and O_2 are zero-functions over domains A_j and A_j' respectively.

By linearity of the involved functions we get $(\sum_{i=1}^{n} v_i \circ (W_{ij} \circ h))(g') = (\sum_{i=1}^{n} v_i \circ W_{ij})(h(g'))$ for all $g' \in A_j'$. It is thus enough to prove the following biimplication

$$\forall g \in A_j \; [(\sum_{i=1}^{n} v_i \circ W_{ij})(g) = O] \Leftrightarrow \forall g' \in A_j' \; [(\sum_{i=1}^{n} v_i \circ W_{ij})(h(g')) = O]$$

$\Rightarrow$ follows directly from the functionality of h, while $\Leftarrow$ follows from pseudosurjectivity of h and from liniarity of the involved functions.

Transformation 2

By linearity of the involved functions we get the following biimplication

$$\sum_{i=1}^{n} v_i \circ W_{ij} = O \;\wedge\; \sum_{i=1}^{n} v_i \circ W_{ik} = O$$

$$\Updownarrow$$

$$\sum_{i=1}^{n} v_i \circ (W_{ij} + W_{ik} \circ h) = O \;\wedge\; \sum_{i=1}^{n} v_i \circ W_{ik} = O$$

Transformation 3

Column j corresponds to an equation, which is always satisfied, and can thus be removed without changing the set of invariants. By transformation 3 we may obtain a matrix with no columns. Such a matrix has as solutions all vectors u which have the correct size and functionality.

Transformation 4

Column j corresponds to an equation, where any solution u must satisfy $u(i) = u(k) \circ g$. When this is the case linearity of the involved functions allow us to combine the two rows without changing the set of equations, and column j can be omitted since the corresponding equation is always satisfied (by the modification of the weightfactors). □

In section 3-5 we consider three examples of coloured Petri nets, and we show how our transformation rules can be used to obtain simple matrices from which invariants can be found directly by inspection.

3. SMALL DATA BASE SYSTEM

Our first example is the network of data bases, described and analysed in [2], section 5. The example has also been used in [1], where it was described in terms of predicate/transition-nets.

In figure 1 the horizontal and vertical lines labelled m0, m1, m2, and m3 indicate four different matrices. The four asterisk's in the horizontal line labelled m0 shows that the incidence-matrix m0 has four columns c1-c4. Analogously the vertical line labelled m0 shows that m0 has eight rows.

						c1	c2	c3	c4	c5	c6
						DBM	DBM	MB	MB	DBM	MB
		m0				*	*	*	*	$T2_1$ c1+c2	
			m1			$T2_1$ c5	*	*	*	*	$T2_2$ c3+c4
				m2			*	$T2_2$ c6	*	*	*
					m3		*		*	$T3_4$	$T4_3$
inactive	DBM	*	*	*	*	-ID	ID	-REC	REC		
waiting	DBM	*	*	*	*	ID	-ID				
performing	DBM	*	*	*	*			REC	-REC		
exclusion	E	*	*	*	*	-ABS	ABS				
unused	MB	*	*	*	*	-MINE	MINE				
sent	MB	*	*	*	$T4_3$	MINE		-ID		MINE	-ID
received	MB	*	*	*	*			ID	-ID		
acknowledged	MB	*	*	*	$T4_3$		-MINE		ID	-MINE	ID
sent, acknowledged	MB			$T4_3$	*		-MINE		ID		

Figure 1. Small data base system, A.

The asterisk's indicate the columns and rows contributing to the individual matrices. Application of transformation rules are indicated by equations. Each equation contains a transformation number (T1-T4) or "LC" (for linear combination). Moreover the equations contain the involved column numbers and the applied functions when these are non-trivial. An equation above (before) an asterisk indicates how the column (row) is created, while an equation below (after) an asterisk indicates how the column (row) is removed. The sequential order of the transformations is shown by subscripts, and this also enables us to distinguish between different applications of the same transformation rule.

From the horizontal lines m0 and m1 it can be seen that column c1 by T2 (transformation 2) has been removed in favour of column c5, which has been created by adding c1 and c2. Analogously c3 is removed in favour of c6 and we have the matrix m2, consisting of c2, c4, c5, c6 and the first eight rows. By T4 c6 is removed and the rows for "sent" and "acknowledged" are combined into a new row. Next T3 allows us to remove c5, and we have the matix m3, also shown in figure 2, together with five solutions to the homogeneous matrix-equation. The five solutions correspond to the five simple invariants found in [2].

m3		c2	c4	m_0	i1	i2	i3	i4	i5
		DBM	MB		DBM	MB	E	DBM	MB
inactive	DBM	ID	REC	ΣDBM	ID				
waiting	DBM	-ID			ID		ABS		MINE
performing	DBM		-REC		ID			ID	
exclusion	E	ABS		1			ID		
unused	MB	MINE		ΣMB		ID			
sent, acknowledged	MB	-MINE	ID			ID			-ID
received	MB		-ID			ID		-REC	-ID

Figure 2. Small data base system, B.

In this paper all applications of T4 will be with g equal to the identity-function or the zero-function. This implies that all weightfactors are identity-functions and they will not be shown. In more complicated examples it is sufficient to show the weightfactors, which differ from the identity-function.

During the transformations it is important to use a systematic notation to show the transformations already performed and the matrix currently obtained. Using the asterisk-notation it is possible to show all transformations and all obtained matrices in a single figure, but often clarity is enhanced by redrawing (and reorganizing) the entire matrix. As a further aid to the eye, columns and rows can be hatched when they are removed.

4. TELEPHONE SYSTEM

Our second example is a telephone system. The coloured Petri net in figure 3 represents the behaviour of individual telephones, as seen by users. The status of a telephone may change from "inactive" to the situation where the receiver has been removed and you hear a "continuous" tone. Next to the situation, where a number u has been dialled and you hear "no tone" until either you get a tone with "short" intervals (indicating that telephone u is already engaged) or you get a tone with "long" intervals (indicating that the bell is "ringing" at telephone u). In the latter situation the receiver may be removed at telephone u and the two telephones are "connected" until the calling telephone returns to "inactive" thereby making telephone u "disconnected".

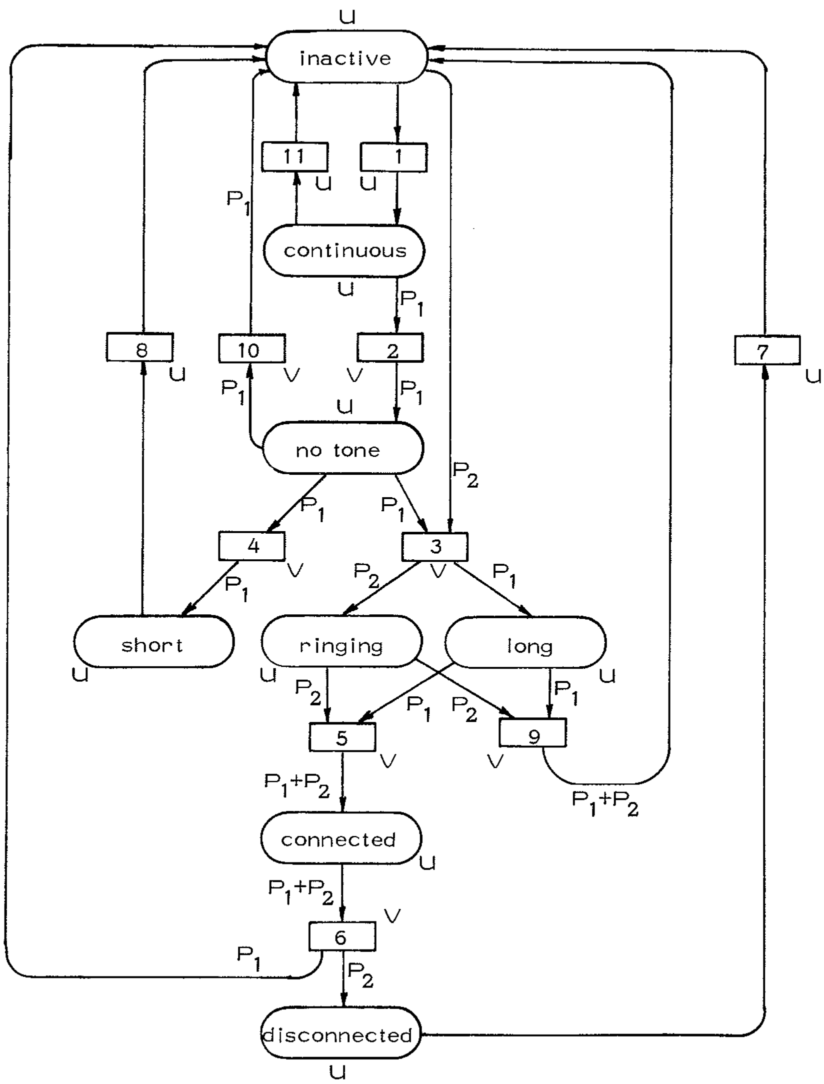

Figure 3. Telephone system, part 1.

U is a set of colours, which represents the different telephone numbers. In $V = U \times U$ the first component represents the calling telephone, while the second component represents the called telephone. P_1 and P_2 are the two projections mapping elements of V in their first and second component, respectively.

The coloured Petri net in figure 3 describes how individual telephones behave, but tells very little about the synchronization between them. This synchronization is described by the coloured Petri net in figure 4, which represents the telephone exchange. Thus the total telephone system consists of figure 3 overlayed by figure 4 (i.e. identification of the transitions with the same number).

Seen from the telephone exchange each telephone may be "free " or "engaged". A pair (u_1, u_2) at "request1 " indicates that u_2 has been dialled from u_1. When u_2 is "free" the call (u_1, u_2) may progress to "request2" and when the receiver is removed at u_2 there is "connexion" between u_1 and u_2.

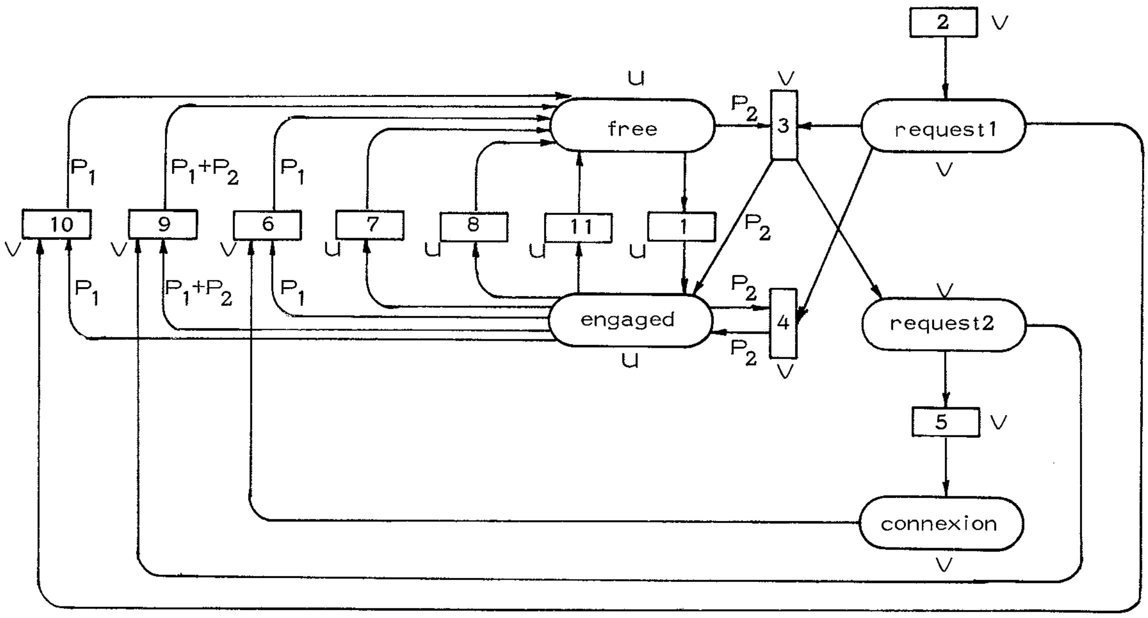

Figure 4. Telephone system, part 2.

The net in figure 4 contains a sidecondition (transition 4 has only concession if the called telephone is "engaged", but the firing of transition 4 does not change the marking of "engaged"). This sidecondition is not reflected in the incidence-matrix shown in figure 5, since no tokens are added or removed by the firing. There is not a bijective correspondence between incidence-matrices and coloured Petri nets with sideconditions, but only between incidence-matrices and coloured Petri nets without sideconditions. Invariants are defined in terms of the incidence-matrix, and thus they can say nothing about system properties relying on the presence of sideconditions. But, as we shall see, there is still a large number of interesting invariants.

Initially m_0(inactive) = m_0(free) = ΣU and all other places are unmarked. The incidence-matrix is indicated by lines m0 in figure 5 and it can be transformed as shown in figures 5, 6 and 7.

					c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13
					U	V	V	V	V	V	U	U	V	V	U	V	U
		m0			*	*	*	*	*	*	*	*	*	*	*	$T2_1$ c2+c4	$T2_2$ c7−c8
			m1		*	$T2_1$ c12	*	*	*	*	$T2_2$ c13	*	*	*	LC_3 −c1	*	*
				m2	*		*	*	*	*		*	*	*		$T4_4$	$T4_4$
inactive	U	*	*	*	−ID		$-P_2$			P_1	ID	ID	P_1+P_2	P_1	ID		
continuous	U	*	*	$T4_4$	ID	$-P_1$									−ID	$-P_1$	
no tone	U	*	*	*		P_1	$-P_1$	$-P_1$						$-P_1$			
long	U	*	*	*			P_1		$-P_1$				$-P_1$				
short	U	*	*	$T4_4$				P_1				−ID				P_1	ID
ringing	U	*	*	*			P_2		$-P_2$				$-P_2$				
connected	U	*	*	*					P_1+P_2	$-P_1-P_2$							
disconnected	U	*	*	$T4_4$						P_2	−ID						−ID
free	U	*	*	*	−ID		$-P_2$			P_1	ID	ID	P_1+P_2	P_1	ID		
engaged	U	*	*	*	ID		P_2			$-P_1$	−ID	−ID	$-P_1-P_2$	$-P_1$	−ID		
request1	V	*	*	*		ID	−ID	−ID						−ID			
request2	V	*	*	*			ID		−ID				−ID				
connexion	V	*	*	*					ID	−ID							
cont,short,disc	U		$T4_4$	*	ID			P_1		P_2		−ID					

Figure 5. Telephone system, A.

		c1	c3	c4	c5	c6	c8	c9	c10	c14	c15
		U	V	V	V	V	U	V	V	V	V
	m2	*	*	*	*	*	*	*	*		
	m3	*	LC_5 c10−c9	*	*	*	LC_8 −c1	LC_7 c5+c6 −c1∘P_2	LC_8 c4− c1∘P_1	$T2_9$ c6+ c1∘P_1	$T2_{10}$ c5+c14
	m4	*		*	$T2_{10}$ c15	$T2_9$ c14				*	*
inactive	U	−ID	$-P_2$			P_1	ID	P_1+P_2	P_1		
cont, short, disc	U	ID		P_1		P_2	−ID			P_1+P_2	P_1+P_2
no tone	U		$-P_1$	$-P_1$					$-P_1$		
long	U		P_1		$-P_1$			$-P_1$			$-P_1$
ringing	U		P_2		$-P_2$			$-P_2$			$-P_2$
connected	U				P_1+P_2	$-P_1-P_2$				$-P_1-P_2$	
free	U	−ID	$-P_2$			P_1	ID	P_1+P_2	P_1		
engaged	U	ID	P_2			$-P_1$	−ID	$-P_1-P_2$	$-P_1$		
request1	V		−ID	−ID					−ID		
request2	V		ID		−ID			−ID			−ID
connexion	V				ID	−ID				−ID	

Figure 6. Telephone system, B.

For the matrix m4 eight solutions to the homogeneous matrix-equation is shown in figure 7, and it is easy to interpret the corresponding invariants in terms of the original system. As an example i7 tells that the "ringing" telephones are exactly those for which a call is waiting at "request2", and i5 tells that a telephone is "connected" iff its number is contained in one of the telephone-pairs having "connexion".

m4		c1	c4	c14	c15	m_0	i1	i2	i3	i4	i5	i6	i7	i8
		U	V	V	V		U	U	U	U	U	U	U	E
inactive	U	−ID				ΣU	ID		ID					
cont, short, disc	U	ID	P_1	P_1+P_2	P_1+P_2		ID							
no tone	U		$-P_1$				ID			−ID				
long	U				$-P_1$		ID					−ID		−ABS
ringing	U				$-P_2$		ID						−ID	ABS
connected	U			$-P_1-P_2$			ID				−ID			
free	U	−ID				ΣU		ID	−ID					
engaged	U	ID						ID						
request1	V		−ID							P_1				
request2	V				−ID							P_1	P_2	
connexion	V			−ID							P_1+P_2			

Figure 7. Telephone system, C.

The coloured Petri net in figures 3 and 4 constitutes a formal model, which allows us to determine even the more subtle properties of the specified telephone system. As an example, we can investigate what happens when a telephone is called by itself, and it can be seen that a "connexion" only can be interrupted by the calling telephone and not by the called telephone. If only an informal description was given, it would be easy to forget about some of these special cases.

Now an important question has to be answered. How did we find the transformations to be used in figure 5 and figure 6? In particular it may seem difficult to know, when it is adequate to apply T2, and to find columns, which are linear combinations of other columns.

In general it is advisable to look for columns, which as far as possible have their non-zero elements in the same rows. As an example c1 and c6 (in figure 6) have four non-zero rows in common. Having made this observation it is rather easy to obtain c14 by means of T2 and the function P_1. Next we have to decide, whether it is c1 or c6, which should be removed in favour of c14. If we, as shown, choose to remove c6 we can use T2 directly. If we choose to remove c1 we must first use T1 to replace c1 by "$c1 \circ P_1$", which then can be removed by T2. Normally we remove the most complicated column (i.e. many non-zero elements or complex functions).

A linear combination will often be established in several steps: Two columns are combined, and some of the elements in the new column are identical to those in an existing column. We record the rows, where the columns differ and search for a third column with non-zero elements in those rows. This process may continue in several steps until we hit the desired column. As an example we may start by combining c1 and c6 according to the equation "$c6 - c1 \circ P_1$". This yields a column with four occurrences of "$P_1 + P_2$", which are present in several other columns. We next notice that this new column is identical to c9 except at the places "long", "ringing", "connected", "request2", and "connexion". But these five places are exactly the non-zero rows for c5, and by adding this column to "$c6 - c1 \circ P_1$" we get c9.

It should be noted that in figure 7 the columns are "independent" in the sense that only the second row has more than one non-zero element.

5. LARGE DATA BASE SYSTEM

Our third example is a network of data bases, similar to that in our first example. But now several managers can be non-passive concurrently and this calls for a more complicated communication discipline. The example has also been used in [1] and [5], where it was described in terms of predicate/transition-nets and synchronization processes respectively. The system is described by the coloured Petri net in figure 8, which is a straightforward translation of the predicate/transition-net given in [1].

The set of colours and the functions are the same as for the small data base system in section 3, except that SEN maps each pair $<s,r> \in MB$ into its first component, which indicates the sender. For convenience some places are shown more than once. The initial marking is $m_0(\text{passive}) = \Sigma DBM$ and $m_0(\text{HOME}) = \Sigma MB$. All other places are unmarked.

Transitions b1, b2 and b3 represent user interaction with one of the managers. A user can request an operation by placing a token on "INTREQ" (b1) and later receive an answer as a token on "DONE" (b2) or "REJECT" (b3).

When manager k finds an internal request on "INTREQ", k passes from "passive" to "active" and informs the other managers by moving the corresponding message buffers from "HOME" to "EXTREQ" (transition 1). Now there are two possibilities:

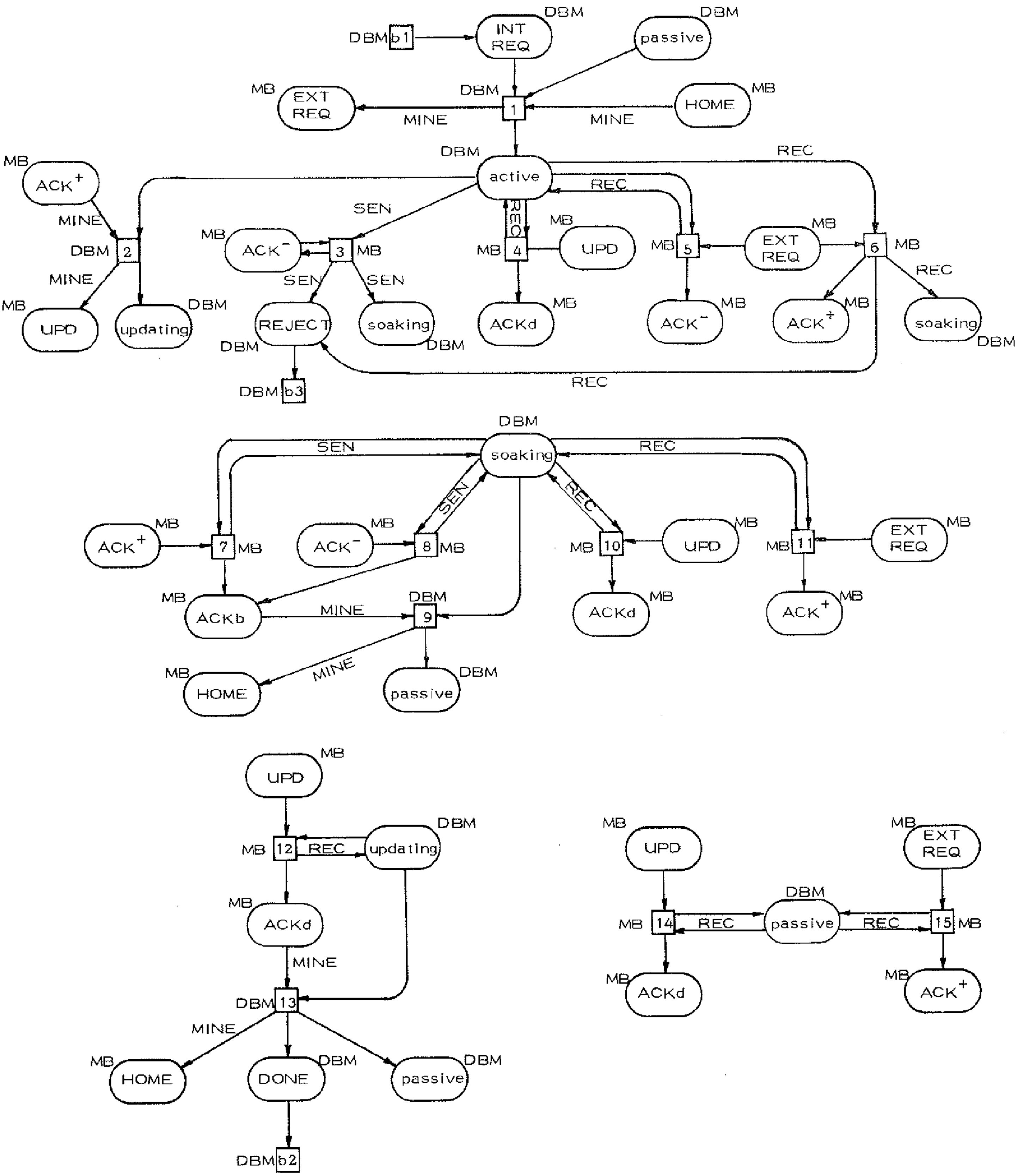

Figure 8. Large data base system.

Manager k gets a positive answer on "ACK$^+$" from all other managers (6, 11, or 15; depending on the state of the manager). Then k performs the update and passes from "active" to "updating" and sends an update-request to the other managers on "UPD" (2). The other managers perform the update and answer k on "ACKd" (4, 10, 12, or 14; depending on the state of the manager). Then k returns the message buffers to "HOME", informs the user at "DONE" and passes from "updating" to "passive" (13).

Manager k receives at least one negative answer on "ACK$^-$" (5). Then k informs the user at "REJECT" and passes from "active" to "soaking" (3). The answers at "ACK$^+$" and "ACK$^-$" are collected at "ACKb" (repeated firing of 7 and 8). Then k returns the message buffers to "HOME" and passes from "soaking" to "passive" (9).

				c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18
				b1	b2	b3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				DBM	DBM	DBM	DBM	DBM	MB	MB	MB	MB	MB	MB	DBM	MB	MB	MB	DBM	MB	MB
		m0		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
			m1	$T4_1$	$T4_2$	$T4_3$	*	*	*	*	*	*	*	*	*	LC_4 c7	*	LC_5 c7	*	LC_6 c7	LC_7 c14
INTREQ	DBM	*	$T4_1$	ID			-ID														
DONE	DBM	*	$T4_2$		-ID														ID		
REJECT	DBM	*	$T4_3$			-ID			SEN			REC									
passive	DBM	*	*				-ID								ID				ID		
active	DBM	*	*				ID	-ID	-SEN			-REC									
soaking	DBM	*	*						SEN			REC			-ID						
updating	DBM	*	*					ID											-ID		
HOME	MB	*	*				-MINE								MINE				MINE		
EXTREQ	MB	*	*				MINE				-ID	-ID					-ID				-ID
ACK$^+$	MB	*	*					-MINE				ID	-ID				ID				ID
ACK$^-$	MB	*	*								ID			-ID							
ACKb	MB	*	*										ID	ID	-MINE						
ACKd	MB	*	*							ID						ID		ID	-MINE	ID	
UPD	MB	*	*					MINE		-ID						-ID		-ID		-ID	

Figure 9. Large data base system, A.

					c4	c5	c6	c7	c8	c9	c10	c11	c12	c14	c16
					DBM	DBM	MB	MB	MB	MB	MB	MB	DBM	MB	DBM
		m1			*	*	*	*	*	*	*	*	*	*	*
			m2		*	*	$T4_8$	$T4_{10}$	$T4_9$	*	$T4_9$	$T4_9$	*	$T4_9$	*
				m3	*	*				$T3_{11}$			*		*
passive	DBM	*	*	*	-ID								ID		ID
active	DBM	*	$T4_8$		ID	-ID	-SEN			-REC					
soaking	DBM	*	$T4_8$				SEN			REC			-ID		
updating	DBM	*	*	*		ID									-ID
HOME	MB	*	*	*	-MINE								MINE		MINE
EXTREQ	MB	*	$T4_9$		MINE				-ID	-ID				-ID	
ACK^+	MB	*	$T4_9$			-MINE				ID	-ID			ID	
ACK^-	MB	*	$T4_9$						ID			-ID			
ACKb	MB	*	$T4_9$								ID	ID	-MINE		
ACKd	MB	*	$T4_{10}$					ID							-MINE
UPD	MB	*	$T4_{10}$			MINE		-ID							
active, soaking	DBM	$T4_8$	*	*	ID	-ID							-ID		
EXTREQ, ACK^+, ACK^-, ACKb	MB	$T4_9$	*	*	MINE	-MINE							-MINE		
ACKd, UPD	MB	$T4_{10}$	*	*		MINE									-MINE

Figure 10. Large data base system, B.

The incidence-matrix can be transformed as shown in figures 9, 10 and 11. For the matrix m4 five solutions to the homogeneous matrix-equation are shown in figure 11, and it is easy to interpret the corresponding invariants in terms of the original system. As an example i4 tells that when a process is "active" or "soaking", all its message buffers are either at "EXTREQ", "ACK^+", "ACK^-", or "ACKb".

It should be noted that the matrix m4 is non-sparse, in the sense that 8 out of 12 elements differ from the zero-function.

		c4	c5	c12	c16	m_0	i1	i2	i3	i4	i5
		DBM	DBM	DBM	DBM		DBM	MB	MB	MB	MB
	m3	*	*	*	*						
	m4	LC_{12} -c12	LC_{13} c12-c16	*	*						
passive	DBM	-ID		ID	ID	ΣDBM	ID		MINE		
active, soaking	DBM	ID	-ID	-ID			ID			MINE	
updating	DBM		ID		-ID		ID				MINE
HOME	MB	-MINE		MINE	MINE	ΣMB		ID	-ID		
EXTREQ, ACK^+, ACK^-, ACKb	MB	MINE	-MINE	-MINE				ID		-ID	
ACKd, UPD	MB		MINE		-MINE			ID			-ID

Figure 11. Large data base system, C.

6. CONCLUSION

We have defined a set of transformation rules, which can be used to transform the incidence-matrices for coloured Petri nets. The transformation rules are sound and independent, but not complete.

Moreover we described three different systems by means of coloured Petri nets. For each system our transformation rules allowed us to obtain a simplified matrix, where it was easy to find solutions for the homogeneous matrix-equation. These solutions were translated to the corresponding invariants and some of these were interpreted in terms of the original coloured Petri net. The degree of simplification achieved by our transformation rules is shown in figure 12.

	Rows		Columns		Elements		Non-zero elements	
	before	after	before	after	before	after	before	after
Small data base system	8	7	4	2	32	14	18	9
Telephone system	13	11	11	4	143	44	52	14
Large data base system	14	6	18	2	252	12	47	8

Figure 12 Size of the matrices before and after application of the transformation rules.

In this paper we have not used transformation 1, and we have only used transformation 4 in the two simple cases, where g is the identity-function or the zero-function. However, in the analysis of larger, more complicated systems, we have found the need for these two transformations.

Our transformation rules can be used to find invariants, but they can also show that certain places cannot be covered by any invariants.

In [4] it has been shown for place/transition-nets that the existence of non-coverable places implies that the net either is not "strongly bounded" or has no "alive" marking. It would be of interest to derive a similar result for coloured Petri nets.

Acknowledgment

This work was promoted by several discussions with Morten Kyng, Mogens Nielsen and Erik Meineche Schmidt.

References

[1] Genrich, H.J. and Lautenbach, K.: The analysis of distributed systems by means of predicate/transition-nets. Semantics of Concurrent Computation, Evian 1979, G. Kahn (ed.), Lecture Notes in Computer Science, vol. 70, Springer Verlag 1979, 123-146.

[2] Jensen, K.: Coloured Petri nets and the invariant-method. DAIMI PB-104, Computer Science Department, Aarhus University, October 1979.

[3] Lautenbach, K.: Liveness in Petri nets. Interner Bericht ISF-75-02. 1, GMD Bonn, July 1975.

[4] Memmi, G. and Roucairol, G.: Linear algebra in net theory. Advanced course on general net theory of processes and systems, Hamburg 1979, H. Fuss (ed.), GMD Bonn 1979. Also to be published by Springer Verlag, 1980.

[5] Milne, G.: Modelling distributed database protocols by synchronization processes. CSR-34-78, Computer Science Department, Edinburgh University, October 1978.