

**Bridging the Gap between Place- and Floyd-Invariants with  
Applications to Preemptive Scheduling.**

Rüdiger Valk.

revised version from

Ajmone Marsan, M., editor, *Application and Theory  
of Petri Nets 1993, Proceedings 14th International  
Conference, Chicago, Illinois, USA*,  
volume 691 of  
*Lecture Notes in Computer Science*,  
pages 433-452. Springer-Verlag, 1993.

Copyright Springer-Verlag Berlin



# **Bridging the Gap Between Place- and Floyd-Invariants with Applications to Preemptive Scheduling**

**Rüdiger Valk**

Fachbereich Informatik, Universität Hamburg  
Vogt-Kölln-Str. 30, D-2000 Hamburg 54  
e-mail: [valk@informatik.uni-hamburg.de](mailto:valk@informatik.uni-hamburg.de)

## **Abstract:**

The notion of linear place-invariants for coloured nets is extended to sums of non-linear functions. The extension applies to such places where all tokens are removed by the occurrence of an output transition. It is shown how this covers the case of variable assignments and invariants in traditional programs. The result helps in understanding the relation of place-invariants of coloured nets in comparison with traditional Floyd-invariants of programs. In the second part the property of token clearing is introduced to the occurrence rule, showing that the results of the first part are still valid. Such types of nets are important for the modelling of fault tolerant applications.

## **Keywords :**

analysis and synthesis, structure and behavior of Petri nets, higher-level net models, coloured Petri nets, program verification, place-invariants, Floyd-invariants, selfmodifying coloured Petri nets

## **1. Introduction**

Invariants have been well studied for ALGOL-like programs [Floyd 67, Hoare 69, Gries 81] and for Petri nets (place-invariants) [Jensen 81]. In formal verification techniques such invariants are used to prove properties that hold invariant over an infinity of possible system states.

On the other hand rather less is known on the relation between these two types of invariants. Floyd-invariants allow formulation of very general properties, but there are no general procedures to compute them. (For a given ALGOL-like program partial correctness is undecidable.) Place-invariants of Petri nets are computable (even in polynomial time for ordinary place/transition nets) but are required to have a linear form.

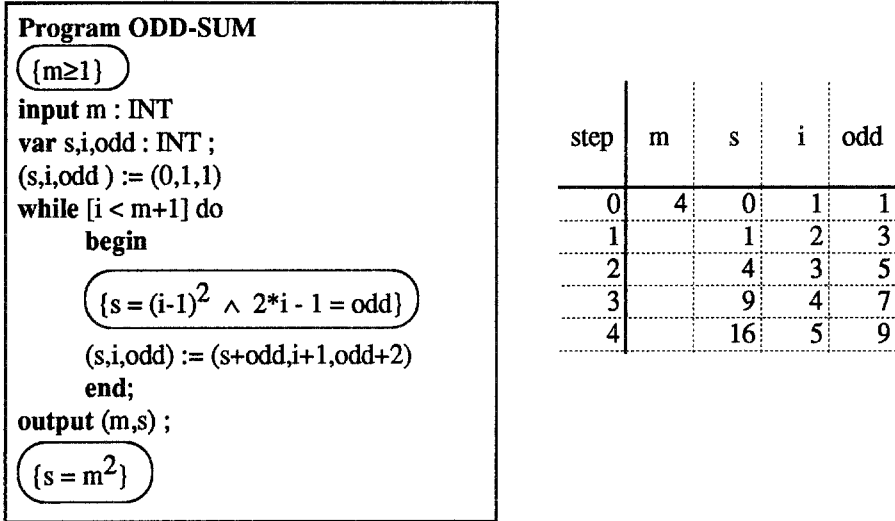


Fig. 1.1: Program ODD-SUM with assertions and a trace

In this paper we will start to bridge the gap between these two formalisms. To begin with, consider the program of fig. 1.1. It computes the square of a given integer  $m$  by successive addition of odd numbers. Traditionally this property is proved by the given pre- and post-conditions and the invariant assertion at the beginning of the while loop. There is also shown a trace including the first four steps.

In section 2 we will formulate this program as a coloured Petri net. Since the given invariant assertion is not linear, it does not fit into the classical place-invariance calculus of coloured nets. But we will prove a general and sufficient condition for computing such invariants from the incidence matrix. The problem has been studied in [Vautherin 85] for a restricted class of nets.

We will give a sufficient local condition for such places, for which the invariant equation contains a non-linear term. This allows us to formulate two theorems that cover (traditional) linear place-invariants and their non-linear extensions and to clarify the different nature of place-invariants of nets on the one hand and of non-linear invariants of programs on the other hand. Moreover by this approach we are able to consider an extended class of coloured nets that is very useful for the modelling of systems requiring the removal of all objects under certain circumstances, e.g. fault tolerant modelling or transitions with preemptive priority. We will show in section 4 that the classical place-invariance calculus as well as the extensions of section 3 still hold for this class of selfmodifying coloured nets.

## 2. Coloured Nets

In this section the technical definition of a coloured net is given. First we recall the notion of a multi-set and of a marking.

**Definition 2.1 :** A *multi-set*  $m$ , over a non-empty set  $S$ , is a function  $m: S \rightarrow \mathbb{IN}$ , sometimes denoted as a formal sum  $\sum_{s \in S} m(s) \cdot s$ .  $S_{MS}$  is the set of all multi-sets over  $S$ . Extending set union to  $S_{MS}$  we define the operation of sum (+) and difference (-).

Since in this paper this can lead to confusion, addition and difference in the sets  $\mathbb{Z}$  of integers and in the subset of non-negative integers  $\mathbb{IN}$  will be denoted by  $+$  and  $-$ , respectively, and  $\Sigma$  is used for general summation.

If  $m, m_1$  and  $m_2$  are multisets over  $S$ , then we define :

$$m_1 + m_2 := \sum_{s \in S} (m_1(s) + m_2(s)) \cdot s$$

$$m_1 \leq m_2 := \Leftrightarrow \forall s \in S : m_1(s) \leq m_2(s),$$

$$\text{and if } m_2 \leq m_1 \text{ then } m_1 - m_2 := \sum_{s \in S} (m_1(s) - m_2(s)) \cdot s.$$

$|m| := \sum_{s \in S} m(s)$  is the *size* of  $m$  and  $\emptyset$  denotes the *empty multi-set* (when  $|m| = 0$ ).

If  $S \subseteq \mathbb{Z}$  is a set of integers, then  $\#m := \sum_{s \in S} m(s) \cdot s$  is the *sum* of  $S$ , in particular  $\#\emptyset = 0$ .

**Definition 2.2** Let  $P$  be a finite set of *places*. To each place  $p \in P$  we associate a *colour set*  $C(p)$ . A *marking*  $M$  is a mapping defined on  $P$  such that  $M(p) \in C(p)_{MS}$ . If  $P = \{p_1, p_2, \dots, p_n\}$  is a totally ordered set, then  $M$  can be written as a vector :

$$\begin{pmatrix} M(p_1) \\ M(p_2) \\ \vdots \\ M(p_n) \end{pmatrix} \in C(p_1)_{MS} \times C(p_2)_{MS} \times \dots \times C(p_n)_{MS} =: C_P = \prod_{p \in P} C(p)_{MS}$$

$C_P$  denotes the set of all possible markings.

To define coloured Petri nets we follow the formalism of [Jensen 87] in the form of a "CP-matrix".

**Definition 2.3** A *coloured Petri net* (CPN)  $N = (P, T, C, \Sigma, I_-, I_+, M_0)$  is given by

- a finite set  $P$  of *places*,
- a finite set  $T$  of *transitions*, disjoint with  $P$ :  $P \cap T = \emptyset$
- a set  $\Sigma$  of *colour sets*,
- a *colour function*  $C : P \cup T \rightarrow \Sigma$ , where
  - $C(p)$  is called the *colour set* of  $p$  and
  - $C(t)$  is said to be the *colour set* (or *occurrence modes*) of  $t$
- $I_+$  and  $I_-$  are the *positive* and *negative incidence functions* on  $P \times T$ :
  - $\forall (p, t) \in P \times T : I_-(p, t), I_+(p, t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L$
  - (As usual, for sets of multisets  $A$  and  $B$   $[A \rightarrow B]_L$  denotes the set of linear functions, cf. [Jensen 87])
- $M_0$  is a marking on  $P$ , called the *initial marking*.

**Definition 2.4** For  $P = \{p_1, p_2, \dots, p_n\}$ ,  $t \in T$  and  $b \in C(t)$  we will use the following notions of vectors  $I_+(-, t)$  and  $I_+(-, t)(b)$ :

$$I_+(-, t) = \begin{pmatrix} I_+(p_1, t) \\ I_+(p_2, t) \\ \vdots \\ I_+(p_n, t) \end{pmatrix} \quad \text{and} \quad I_+(-, t)(b) = \begin{pmatrix} I_+(p_1, t)(b) \\ I_+(p_2, t)(b) \\ \vdots \\ I_+(p_n, t)(b) \end{pmatrix}$$

$I_+(-, t)(b)$  is extended to hold for multisets  $b \in C(t)_{MS}$ . The corresponding definitions for  $I_-(-, t)$  and  $I_-(-, t)(b)$  are omitted.

**Example 2.1.** Fig. 2.1 gives the incidence functions of the CPN *ODD\_SUM* in the form of a common matrix. The values of  $I_-$  are placed in left upper corners of the entries and are represented in negative form. The corresponding lower right corners give the values of  $I_+$ . The colours of places and transitions are also given. The entries in the incidence matrix refer to the corresponding projections, e.g.:  $I_+(p_1, t_2) = s+odd = pr_{C(p_2)}(C(t_2)) + pr_{C(p_{odd})}(C(t_2))$ , where  $pr_{C(p_2)}(C(t_2))$  is the projection on the first component of  $C(t_2)$ , i.e. on  $C(p_2)$ . To avoid confusion, the function  $s+odd : C(t_2) \rightarrow C(p_1)$  is explicitly defined by  $(s+odd)(s, i, odd) := s + odd$ .

The initial marking is given in the first column :

$$M_0 = (M_0(p_1), M_0(p_2), M_0(p_i), M_0(p_m), M_0(p_{odd}), M_0(p_3)) = (1^0, \emptyset, 1^1, 1^4, 1^1, \emptyset)$$



### 3. Extended place-invariants

In the CPN  $N = ODD\_SUM$  the following equations hold for all reachable markings:

$$(A1) \quad \forall M \in R(N) : |M(p1)| + |M(p2)| + |M(p3)| = 1$$

$$(A2) \quad \forall M \in R(N) : |M(pi)| = 1$$

$$(A3) \quad \forall M \in R(N) : |M(podd)| = 1$$

$$(A4) \quad \forall M \in R(N) : \#M(p1) + \#M(p2) + \#M(p3) = (\#M(pi) - 1)^2$$

$$(A5) \quad \forall M \in R(N) : \#M(podd) = 2 * \#M(pi) - 1$$

The first three equations count the number of objects in the places. The fourth equation relates the values of tokens in  $p1$ ,  $p2$  or  $p3$  (if any) to the token in the place  $pi$ . The fifth equation says that the place  $podd$  always contains the  $i$ -th odd number. Invariant equations (A4) and (A5) correspond to the loop invariant of the program ODD-SUM.

As usual these invariants are used to prove the partial correctness of the net: in a terminating marking  $M_E$  there is an object  $q$  in  $p3$  (and by (A1) only one) and from the guard  $[i=m+1]$  of  $t3$  we have  $M_E(pm) = M_E(pi) - 1$ . By equation (A1)  $p1$  and  $p2$  are empty, hence by (A4) :  $q = \#M_E(p3) = (\#M_E(pi) - 1)^2 = (M_E(pm))^2 = m^2$  i.e. the net computes the square of the "input"  $m$ .

Ordinary place-invariants can be deduced directly from the incidence functions. This is a very important property since they can be checked by inspection of the static description of the net. Hence it is not necessary to compute the reachability set, which is very complex in most cases. This property strongly depends on the linearity of ordinary place-invariants and on an invariance of weighted flow for each transition.

Different to this situation equation (A4) is not linear as in numerous similar cases. Therefore the technique mentioned does not apply in general. However, case studies have shown that for many coloured nets modelling classical sequential programs, the invariance of weighted flow is fulfilled.

To prove the invariant equation (A4) by induction, the fifth equation (A5) is also used, as will be shown at the end of this section. Opposite to this fact, classical place invariants can be independently verified. In the following, we will give a number of sufficient conditions for also including non-linear equations into the invariance calculus of coloured Petri nets.

**Definition 3.1** A pair  $(p,t) \in P \times T$  of a CPN is a *clearing arc* if

$\forall M \in R(N) \forall b \in C(t) : M[(t,b) > \Rightarrow I_-(p,t)(b) = M(p)$ . A place  $p \in P$  is a *clearing place* if for all  $t \in T$  and  $c \in C(t)$  either  $I_-(p,t)(b) = I_+(p,t)(b) = \emptyset$  or  $(p,t)$  is a clearing arc. If  $p$  does not have this property, it is called *non-clearing*.

By this definition an occurring transition removes all tokens from input clearing places. Obviously, this definition is not "static" in the sense mentioned above, but can be easily checked by formal computation in many examples, e.g. in the net *ODD\_SUM* place  $p_i$  obviously has this property.

Equations (A1) to (A5) have the form  $\sum_{p \in P} W_p(M(p)) = c$  where  $W_p(M(p))$  has values in an additive group  $\mathbb{A}$  and  $c \in \mathbb{A}$  is a constant.

**Definition 3.2** Let  $(\mathbb{A}, +)$  be an (additively written) commutative group. For a place  $p$  a function  $W_p : C(p)_{MS} \rightarrow \mathbb{A}$  is linear if  $W_p(\emptyset) = 0$  and

$$W_p(x+y) = W_p(x) + W_p(y) \text{ for all } x, y \in C(p)_{MS}.$$

For a marking  $M$  the *weight function*  $W_P : C_P \rightarrow \mathbb{A}$  is defined by  $W_P(M) :=$

$$\sum_{p \in P} W_p(M(p)).$$

" $W_P(M) = c$ " for a constant  $c \in \mathbb{A}$  is called a *weighted equation*.

The weighted equation is called an *invariant equation* of a CPN  $N$  if  $W_P(M) = c$  for all reachable markings  $M \in R(N)$ .

**Remark 3.1:**  $|M|$  and  $\#M$ , as introduced in definition 2.1, are examples of linear functions. Note that the definition of a clearing arc (definition 3.1) also covers the case where  $I_-(p,t)(b) = M(p) = \emptyset$ .

**Theorem 3.3** Let  $W_P$  be a weight function for a CPN  $N = (P, T, C, \Sigma, I_-, I_+, M_0)$ , such that  $W_p$  is linear on all non-clearing places  $p \in P$ .

If  $W_P(M_0) = c$  and  $W_P[I_+(-,t)(b)] = W_P[I_-(-,t)(b)]$  for all  $t \in T$  and  $b \in C(t)$ , then  $W_P(M) = c$  is an invariant equation of  $N$ . The theorem remains true if " $=c$ " is replaced by " $\leq c$ " or " $\geq c$ ".

**Proof** (by induction on the reachability set  $R(N)$ ) :

a)  $W_P(M_0) = c$  (or  $\leq c, \geq c$ , resp.) by assumption.

b) Let  $W_P(M) = c$  (or  $\leq c, \geq c$ , resp.) and  $M[(t,b) > M'$  be an occurrence of  $X = (t,b)$  such that *not*  $I_-(p,t)(b) = I_+(p,t)(b) = \emptyset$ . We have to prove  $W_P(M') = c$ .

$$\text{In fact, } W_P(M') = \sum_{p \in P} W_p[M'(p)] = \sum_{p \in P} W_p[M(p) - I_-(p,t)(b) + I_+(p,t)(b)] \quad (I)$$

b1) If  $p$  is a non-clearing place, then  $W_p$  is linear :

$$W_p[ M(p) - I_-(p,t)(b) + I_+(p,t)(b) ] = W_p[M(p)] - W_p[ I_-(p,t)(b) ] + W_p[ I_+(p,t)(b) ]$$

b2) If  $p$  is a clearing place, then since  $M[(t,b) > 0]$  we have :

$$M(p) - I_-(p,t)(b) = 0 \tag{*}$$

and  $W_p[M(p)] - W_p[I_-(p,t)(b)] = 0 \tag{**}$

By calculating

$$W_p[ M(p) - I_-(p,t)(b) + I_+(p,t)(b) ] = \text{(by (*))}$$

$$W_p[ I_+(p,t)(b) ] = \text{(by (**))}$$

$W_p[M(p)] - W_p[ I_-(p,t)(b) ] + W_p[ I_+(p,t)(b) ]$  we obtain the same result as in case b1). Now we are able to continue with line (I) :

$$W_P(M) = \sum_{p \in P} W_p[M(p)] = \sum_{p \in P} W_p[ M(p) - I_-(p,t)(b) + I_+(p,t)(b) ] = \text{(by b1 and b2)}$$

$$\sum_{p \in P} (W_p[ M(p) ] - W_p[ I_-(p,t)(b) ] + W_p[ I_+(p,t)(b) ]) = \text{(by commutativity in } \mathbb{A} \text{)}$$

$$\sum_{p \in P} W_p[ M(p) ] - \sum_{p \in P} W_p[ I_-(p,t)(b) ] + \sum_{p \in P} W_p[ I_+(p,t)(b) ] = \text{(by definition)}$$

$$W_P(M) - W_P(I_-(\cdot,t)(b)) + W_P(I_+(\cdot,t)(b)) = W_P(M) = c \text{ (or } \leq c, \geq c, \text{ resp.)}$$

(by assumption of  $W_p[I_+(\cdot,t)(b)] = W_p[I_-(\cdot,t)(b)]$  and by induction). □

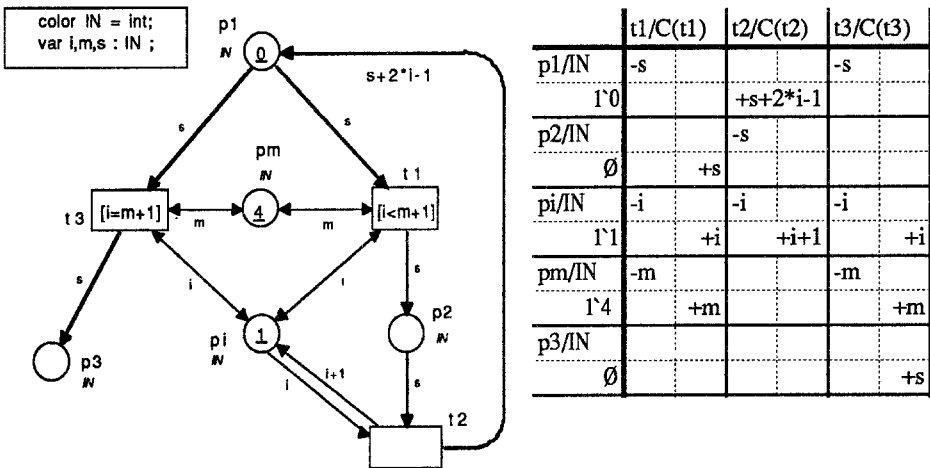


Fig. 3.1: Coloured net *ODD\_SUM-1* with incidence functions  $I_+$  and  $I_-$  and colours.

**Example 3.1:** The theorem does *not* allow the proof of the most interesting invariant (A4). This is however the case for the following, slightly modified CPN *ODD\_SUM-1*, where the place *podd* is omitted and the function  $I_+(p1,t2)$  is replaced by  $s+2*i-1$ . Sin-

ce (A4) has the form  $\forall M \in R(N) : \#M(p1) + \#M(p2) + \#M(p3) - (\#M(pi) - 1)^2 = 0$ , the functions  $W_p$  are defined as follows :  $W_{p1}(a) = W_{p2}(a) = W_{p3}(a) = \#a$ ,  $W_{pi}(a) = -(\#a - 1)^2$ ,  $W_{pm}(a) = 0$ .

Condition  $W_P[I_+(-, t)(b)] = W_P[I_-(-, t)(b)]$  of the theorem is satisfied for  $t=t2$  and  $b = (s, i) \in IN^2$  (we have to look to the nonempty entries of  $I_-$  and  $I_+$  only) :

$$\text{left-hand side : } W_{p1}[I_+(p1, t2)(s, i)] + W_{pi}[I_+(pi, t2)(s, i)] = \\ W_{p1}[s + 2*i - 1] + W_{pi}[i + 1] = s + 2*i - 1 - ((i + 1) - 1)^2 = s + 2*i - 1 - i^2.$$

*right-hand side :*  $W_{p2}[I_-(p2, t2)(s, i)] + W_{pi}[I_-(pi, t2)(s, i)] = W_{p2}[s] + W_{pi}[i] = s - (i - 1)^2 = s - i^2 + 2*i - 1$ . The analogous calculations for  $t1$  and  $t3$  are simple, since  $pi$  is only trivially involved. End of Example 3.1.

**Example 3.2:** Let us return to the example 2.1 with the CPN *ODD\_SUM*. The analogous calculation as in the preceding example yields (observe that  $W_{podd}(a)=0$ ):

$$\text{left-hand side : } W_P[I_+(-, t2)(b)] = W_{p1}[I_+(p1, t2)(s, i)] + W_{pi}[I_+(pi, t2)(s, i)] = \\ W_{p1}[s + odd] + W_{pi}[i + 1] = s + odd - ((i + 1) - 1)^2 = s + odd - i^2.$$

*right-hand side :*  $W_P[I_-(-, t2)(b)] = W_{p2}[I_-(p2, t2)(s, i)] + W_{pi}[I_-(pi, t2)(s, i)] = W_{p2}[s] + W_{pi}[i] = s - (i - 1)^2 = s - i^2 + 2*i - 1$ .

Hence for  $b = (s, i, odd)$  the difference  $W_P[I_+(-, t2)(b)] - W_P[I_-(-, t2)(b)] = odd - 2*i + 1$  is not zero, and the condition  $W_P[I_+(-, t2)(b)] = W_P[I_-(-, t2)(b)]$  of the theorem is *not* satisfied. However, in this example we have the invariant equation

$$(A5) \quad \forall M \in R(N) : \#M(podd) - 2* \#M(pi) + 1 = 0$$

In all reachable markings by (A2) and (A3) there is exactly one token in  $pi$  and  $podd$ , having values  $i$  and  $odd$ , respectively. By (A5) they satisfy  $odd - 2*i + 1 = 0$ , hence also (A4) is an invariant equation. On the other hand, (A5) can be verified by application of theorem 3.3. This is done by defining  $W_{pi}(a) := 1 - 2*\#a$ ,  $W_{podd}(a) := \#a$  and  $W_p(a) = 0$  for  $p \notin \{pi, podd\}$  and the calculation (e.g. for  $t = t2$  and  $b = (s, i, odd)$ ):  $W_P[I_+(-, t2)(b)] - W_P[I_-(-, t2)(b)] = W_{pi}[i + 1] + W_{podd}[odd + 2] - (W_{pi}[i] + W_{podd}[odd]) = 1 - 2*(i + 1) + odd + 2 - ((1 - 2*i) + odd) = 0$ . End of Example 3.2

This example is of particular interest, since different from the case of ordinary and linear place-invariants, some equations cannot be verified by the incidence function independently from other equations. Note that (A4) can be transformed by substi-

tuting #M(pi) by its value from (A5). Then the problem would not occur. Such transformations, however, require special insight, and may lead to clumsy equations.

When formalizing the method we will use a property that is frequently satisfied by coloured Petri-nets arising from practical applications, i.e. the colour sets C(t) of transitions are subsets of products of place colours (c.f. example 2.1).

**Theorem 3.4** Let  $N = (P, T, C, \Sigma, I_-, I_+, M_0)$  be a CPN, where all transition colours C(t) are subsets of Cartesian products of some place colours :

$$C(t) \subseteq C(p_{i_1}) \times \dots \times C(p_{i_m}), \text{ and let} \quad \begin{array}{ll} \text{(C1)} & W_P^1(M) = c_1 \\ \text{(C2)} & W_P^2(M) = c_2 \\ & \dots\dots \\ \text{(Cr)} & W_P^r(M) = c_r \end{array}$$

be a set of weight functions ( $r \geq 2$ ) such that for each equation (Ci) ( $1 \leq i \leq r$ ) the following holds :

- $W_P^i$  is linear on all non-clearing places,
- $W_P^i(M) = c_i$  for the initial marking  $M = M_0$ , and
- either  $\forall t \in T \forall b \in C(t) : W_P^i[I_+(-, t)(b)] - W_P^i[I_-(-, t)(b)] = 0$   
 or  $[\forall t \in T \forall b \in C(t) : W_P^i[I_+(-, t)(b)] - W_P^i[I_-(-, t)(b)] = q(b) \neq 0$   
 and  $b = (x_1, \dots, x_v)$   
 and all  $x_i$  are inscriptions of clearing arcs  
 and  $\forall M \in R(N) : q(b) = W_P^j(M) = 0$  for some  $j$  with  $1 \leq j < i$ ]

Then (C1) ... (Cr) are invariant equations of N.

Technical remark :  $q(b) = W_P^j(M) = 0$  is supposed to be the zero mapping, and not a constant.

**Proof :** Theorem 3.4 follows by successive verification of (C1), (C2), .....

If  $\forall t \in T \forall b \in C(t) : W_P^i[I_+(-, t)(b)] - W_P^i[I_-(-, t)(b)] = 0$  then (Ci) holds by theorem 3.3. If  $\forall t \in T \forall b \in C(t) : W_P^i[I_+(-, t)(b)] - W_P^i[I_-(-, t)(b)] = q(b) \neq 0$ , then  $q(b) = W_P^j(M) = 0$  for some  $j$  with  $1 \leq j < i$  and theorem 3.3. can also be applied. This is justified by the condition that all  $x_i$  are inscriptions of clearing arcs.  $b = (x_1, \dots, x_v)$  is evaluated in such a way that  $x_u$  has the value of the token in the input place  $p_u$  of the corresponding arc. Then  $q(b) = q(x_1, \dots, x_v) = W_{p_u}^j(M) = 0$ . (The evaluation may

be a formal one, as in the following example). □

**Example 3.3:** Theorem 3.4 is illustrated by the equations of example 3.2 : take (A5) as (C1) and (A4) as (C2). For  $t = t_2$  and  $b = (s, i, \text{odd}) \in C(t_2)$  we have  $q(b) = \text{odd} - 2 * i + 1$ . By (C1) = (A5) we have  $\#M(\text{podd}) - 2 * \#M(\text{pi}) + 1 = 0$  and  $\text{odd} - 2 * i + 1 = 0$ .

End of Example 3.3

**Remark 3.2:** The net *ODD\_SUM* can be transformed in such a way that concurrent behavior is possible. By changing the initial marking of place  $p_1$  (see fig. 2.1) to 3'0 three objects are created, each computing part of the sum to be calculated. Finally we might have the end marking  $(M(p_1), M(p_2), M(p_m), M(p_i), M(\text{podd}), M(p_3)) = (\emptyset, \emptyset, 1^4, 1^5, 1^7, 1^1 + 1^8 + 1^7)$ . This could be interpreted as three processes, each of which is computing a part of  $4^2 = 16$  by sharing the places  $p_m$ ,  $p_i$  and  $\text{podd}$ . Invariant equations (A4) and (A5) are still valid, since the operator # collects the sum of all objects in a place. The end marking above stems from a trace where the first token received the value of the first odd number (1), the second the sum of the second (3) and the third (5) and the third token the value of the fourth odd number (7). (Obviously, the loop exit control has to be changed for this extension.)

## 4. Clearing places by occurrence rule

The property of clearing places is frequently desirable in practical system modeling by coloured Petri nets. Often a situation appears, where all messages of a channel have to be removed, but where the actual number of messages is unknown. Similar situations occur if an indefinite number of resources needs an update. Furthermore, a similar case appears in modelling fault tolerant applications. In such cases an ordinary set of tasks or processes is stopped by a super-task with maximal priority. Then by some preemptive scheduling rule all tokens representing ordinary tasks are substituted by the prioritized process.

In fig. 4.1 a node of a queuing network (called elementary waiting system) contains a waiting pool where tasks  $a_i$  are waiting to be processed by the functional unit

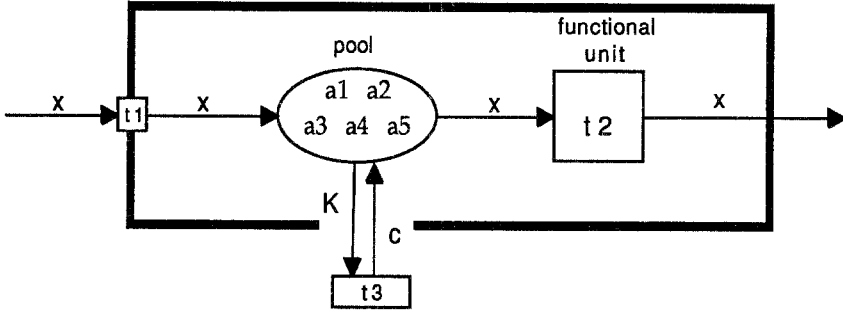


Fig. 4.1 : Elementary waiting system with preemptive scheduling

t2. In the case of system failure the prioritized super-task *c* enters the pool and removes all tasks by a preemptive scheduling rule. Then by the arc *K* all the tasks in the pool have to be removed.

Fig. 4.2 shows a set of tasks or processes  $a_1 \dots, a_{100}$  in a processor pipeline. In case of a system failure all processes in the pools  $p_1$  to  $p_4$  are set back to their initial position by the recovery transition  $t_5$ . This is done by arc expressions ' $p_i$ ', which are evaluated to the multi-sets of the places  $p_i$  in the actual marking. The arc from  $t_5$  to  $p_0$  adds the sum of all these multi-sets to the initial place  $p_0$ .

The same system could be modelled as an ordinary coloured net by considering as tokens the multi-sets of tasks in the pools  $p_i$  (see [Valk 93]). Such a modelling, however, is felt to be rather artificial, since individual objects (tasks) are not represented by individual tokens.

Arcs like  $(p_1, t_5)$  in fig. 4.2 are clearing arcs as introduced in definition 3.1. But

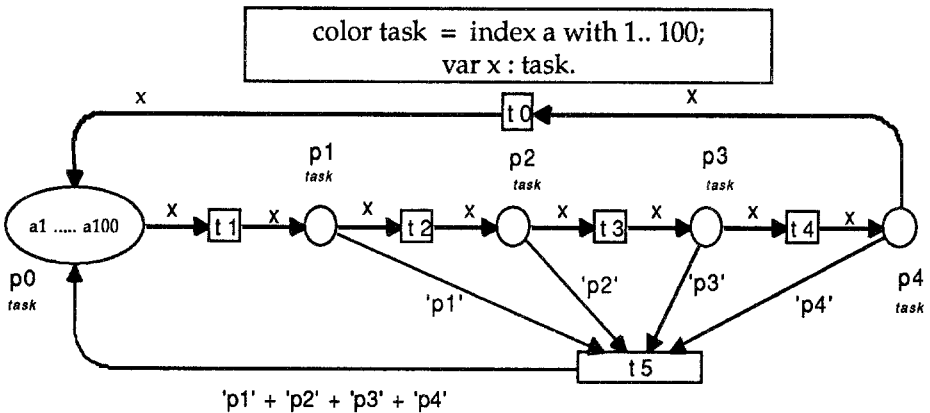


Fig. 4.2 : A process line with recovery

here this property is no longer a dynamical changing one, but is permanently holding by extension of the occurrence rule. This is an instance of selfmodifying coloured nets, which have been studied for the subclass of place/transition nets [Valk 78, Valk 83].

In a selfmodifying net the value of arc inscriptions are allowed to depend on the token content of some specified places. Hence, the structure of arcs between places and transitions is no longer time independent, but may change with the behavior of the net. This explains the name "selfmodifying net". Wellknown extensions like inhibitor arcs, priority transitions and reset arcs are subclasses of selfmodifying nets.

Although in this paper only the feature of clearing places will be used, it is easier in the definition to use selfmodifying coloured nets in general. Selfmodifying coloured nets (smCPN) differ from ordinary coloured nets (CPN, def. 2.3) only in the definition of the incidence functions which may depend on the actual marking. Later on, in definition 4.3, we will introduce a notation for expressing this dependence by arc inscriptions in the graphical representation of selfmodifying coloured nets.

**Definition 4.1** A *selfmodifying coloured Petri net* (smCPN)

$N = (P, T, C, \Sigma, I_{M_-}, I_{M_+}, M_0)$  is given by

- a finite set  $P$  of *places*,
- a finite set  $T$  of *transitions*, disjoint with  $P$  :  $P \cap T = \emptyset$
- a set  $\Sigma$  of *colour sets*,
- a *colour function*  $C : P \cup T \rightarrow \Sigma$ , where

$C(p)$  is called the *colour set* of  $p$  and

$C(t)$  is said to be the *colour set* (or *occurrence modes*) of  $t$

- for every marking  $M \in C_P$   $I_{M_-}$  and  $I_{M_+}$  are the *positive* and *negative incidence functions* on  $P \times T$  :

$$\forall (p, t) \in P \times T : I_{M_-}(p, t), I_{M_+}(p, t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L$$

- $M_0$  is a marking on  $P$ , called *initial marking*.

**Definition 4.2** • A transition  $t$  is *enabled* at a marking  $M$  with binding (colour)

$b \in C(t)$  if  $M \geq I_{M_-} * X$  and  $X = (t, b)$ .

(Again,  $I_{M_-} * (t, b)$  is the vector  $I_{M_-}(-, t)(b)$ .) We write  $M [X >$  in this case.

- If  $t$  is enabled in  $M$ , then the follower marking relation  $M[t, b > M'$  or  $M[X > M'$  is defined by the *follower marking*  $M' := (M - I_{M^-} * X) + I_{M^+} * X$   
(  $I_{M^+} * X = I_{M^+} * (t, b)$  is defined in the same way as  $I_{M^-} * (t, b)$  )
- As usual, the relation  $M[X > M'$  is extended to words  $u$  over  $Z = \{(t, b) | t \in T, b \in C(t)\}$  and the reachability set is  $R(N) := \{ M | \exists u \in Z^* : M_0[u > M] \} \subseteq C_P$ .

While the definition of the "CP-matrix" of smCPN requires only few changes, for the definitions of concrete marking-dependent functions some notations are useful.

**Definition 4.3** • For a multi-set  $m : S \rightarrow \mathbb{N}$  and  $s \in S$  let  $m[s] := m(s)$  be the sub-multi-set of occurrences of  $s$ .

- Let  $p$  be a place with colour set  $C(p) = (X_1)_{MS} \times (X_2)_{MS} \times \dots \times (X_n)_{MS}$  and  $M$  a marking  $M \in C_P$ . Then for  $x \in X_i$  we define

$$M[p(x|i)] := \{ (m_1, m_2, \dots, m_n) \in (X_1)_{MS} \times (X_2)_{MS} \times \dots \times (X_n)_{MS} \mid \\ \exists (q_1, \dots, q_n) \in M(p) : q_j = m_j \text{ for } j \neq i \text{ and } m_i = q_i[x] \neq \emptyset \}.$$

( $M(p(x|i))$  is the set of all tuples in  $M(p)$  having a (positive) multiple of  $x$  as  $i$ -th component.)

- For  $n=1$  we write  $M[p(x)] := M[p(x|1)]$ .

As an example, let  $X_1 := \{a_1, a_2\}$ ,  $X_2 := \{f_1, f_2\}$  and  $M(p) := \{(2a_1, 3f_2), (1a_2, 2f_1), (1a_3, 2f_1 + 1f_2)\}$ . Then  $M[p(f_2|2)] = \{(2a_1, 3f_2), (1a_3, 1f_2)\}$

**Definition 4.4** When using definition 4.3 as arc inscriptions of smCPN in graphical form or as entries of the incidence functions, the explicit reference to the actual marking  $M$  is omitted: i.e. instead of  $M[p(x|i)]$  we use  $p(x|i)$ , since there is no confusion. In a similar way, we will use  $p()$  instead of  $M(p)$ .

(This definition replaces the notion of  $K$  in fig. 4.1 and of ' $\pi_i$ ' in fig. 4.2. The brackets distinguish  $p()$  from the name of the place  $p \in P$ .)

**Example 4.1:** Fig. 4.4 shows an instance of the problem of readers and writers as a smCPN  $\mathcal{R}_P\mathcal{W}$  in graphical form.

There are a set of 5 tasks  $A = \{a_1, a_2, \dots, a_5\}$  and two files  $F = \{f_1, f_2\}$ . Tasks entering the critical region as writers (transition  $t_3$ ) have priority. They start immediately with reading and writing a file  $y=f_i$ . By preemptive scheduling all tasks reading the

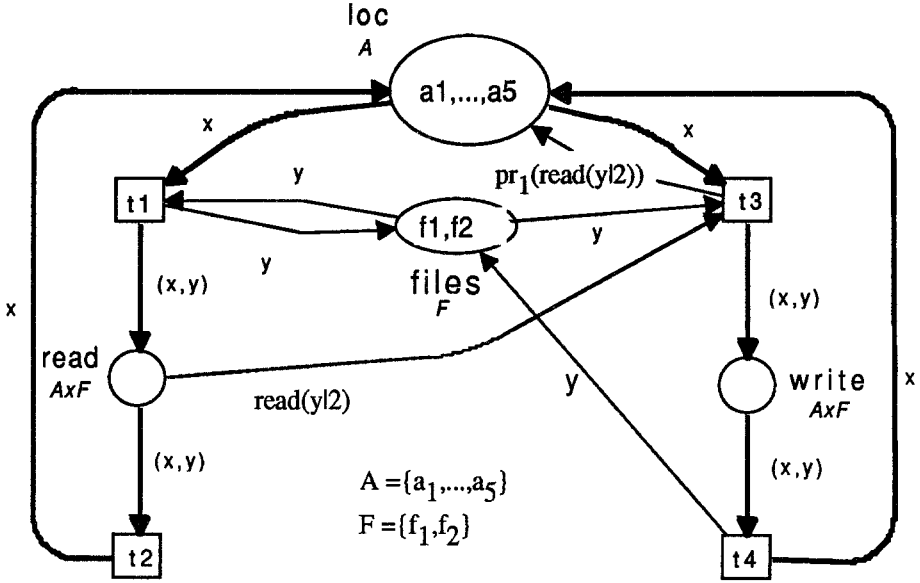


Fig. 4.3:  $Rpw$ : readers and preemptive writers

same file in the place "read" are interrupted and set back to their initial state in  $loc$ . Later they can try to repeat their action of reading. For instance,  $I_{M_+}(loc, t_3) = pr_1(read(y|2)) = \{ a_j | (a_j, f_j) \in M(read) \wedge f_j=y \}$  is the set of all tasks  $a_j \in A$  that are reading the file  $y$  in "read". (Recall that by definition 4.4  $pr_1(read(y|2))$  stands for  $pr_1(M[read(y|2)])$ ).

The smCPN  $Rpw$  satisfies the following place-invariant equations (B1), (B2) and the inequality (B3):

$$(B1) \quad \forall M \in R(\mathbb{N}) : M(loc) + pr_1(M(read)) + pr_1(M(write)) = A$$

$$(B2) \quad \forall M \in R(\mathbb{N}) : M(files) + pr_2(M(write)) = F$$

$$(B3) \quad \forall M \in R(\mathbb{N}) : pr_2(M(read)) + 5 pr_2(M(write)) \leq 5 F$$

It is easy to see that the following property can be verified by (B3): no task is reading a file  $f$  in the place "read" that is also used in the place "write" (in fact, ' $f$ ' in  $pr_2(M(write))$  implies  $pr_2(M[read(f|2)]) = \emptyset$ ).

To verify (B1) - (B3) we reformulate the theorem from section 3.

**Definition 4.5** A pair  $(p, t) \in P \times T$  of a smCPN is a *clearing arc* if

$\forall M \in R(\mathbb{N}) \forall b \in C(t) : M[(t, b)] \Rightarrow I_{M_-}(p, t)(b) = M(p)$ . A place  $p \in P$  is a *clearing place* if for all  $t \in T$  and  $b \in C(t)$  either  $I_{M_-}(p, t)(b) = I_{M_+}(p, t)(b) = \emptyset$  or  $(p, t)$  is a

clearing arc. If  $p$  does not have this property, it is called *non-clearing*.

**Theorem 4.6** Let  $W_P$  be a weight function for a smCPN

$N = (P, T, C, \Sigma, I_{M-}, I_{M+}, M_0)$ , such that  $W_P$  is linear on all non-clearing places.

If  $W_P(M_0) = c$  and  $W_P[I_{M+}(-,t)(b)] = W_P[I_{M-}(-,t)(b)]$  for all  $t \in T$  and  $b \in C(t)$ , then  $W_P(M) = c$  is an invariant equation of  $N$ . The theorem remains true if " $=c$ " is replaced by " $\leq c$ " or " $\geq c$ ".

**Proof:** Replace  $I_-$  by  $I_{M-}$  and  $I_+$  by  $I_{M+}$  in the proof of theorem 3.3.  $\square$

Frequently, as in our example for a weight function  $W_P: \mathcal{C}_P \rightarrow \mathbb{A}$ , the set  $\mathbb{A}$  is a set of multi-sets. Since in our formalism  $\mathbb{A}$  is supposed to be a commutative group, we extend multi-sets to "extended multi-sets", which have coefficients in  $\mathbb{Z}$ .

**Definition 4.7** An *extended multi-set*  $m$ , over a non-empty set  $S$ , is a function  $m: S \rightarrow \mathbb{Z}$  into the set of integers  $\mathbb{Z}$ .  $\mathcal{S}_{MS}$  is the set of all extended multi-sets over  $S$ . The operation  $+$  and the relation  $\leq$  are defined as in definition 2.1. Moreover  $m_1 - m_2 := \sum_{s \in S} (m_1(s) - m_2(s)) \cdot s$  is defined without the precondition  $m_2 \leq m_1$ .

**Example 4.1 (cont.):** Applying the preceding theorem to the invariant equation

$$(B1) \quad \forall M \in R(N) : M(\text{loc}) + pr_1(M(\text{read})) + pr_1(M(\text{write})) = A$$

of the smCPN of example 4.1, the weight function  $W_P: \mathcal{C}_P \rightarrow \mathbb{A}$  is selected as follows :  $\mathbb{A}$  is the set of all extended multisets  $\mathbb{A}_{MS}$  over  $A$ .

$$W_{\text{loc}}(m) := m, W_{\text{read}}(m) = W_{\text{write}}(m) := pr_1(m) \text{ and } W_{\text{files}}(m) := \emptyset.$$

Then  $W_P(M_0) = A$  is satisfied and the equation  $W_P[I_{M+}(-,t)(b)] =$

$W_P[I_{M-}(-,t)(b)]$  is calculated for (e.g.)  $t = t_3$  and  $b = (a, f)$  as follows. (The incidence matrices are not given here. They are constructed in the same way as in section 2.)

$$\begin{aligned} \text{left-hand side :} \quad & W_{\text{loc}}[I_{M+}(\text{loc}, t_3)(a, f)] \div W_{\text{read}}[I_{M+}(\text{read}, t_3)(a, f)] + \\ & W_{\text{write}}[I_{M+}(\text{write}, t_3)(a, f)] \div W_{\text{files}}[I_{M+}(\text{files}, t_3)(a, f)] = pr_1(\text{read}(fl2)) \div \emptyset + \\ & pr_1((a, f)) + \emptyset = pr_1(\text{read}(fl2)) \div 1 \cdot a. \end{aligned}$$

$$\begin{aligned} \text{right-hand side :} \quad & W_{\text{loc}}[I_{M-}(\text{loc}, t_3)(a, f)] \div W_{\text{read}}[I_{M-}(\text{read}, t_3)(a, f)] + \\ & W_{\text{write}}[I_{M-}(\text{write}, t_3)(a, f)] \div W_{\text{files}}[I_{M-}(\text{files}, t_3)(a, f)] = 1 \cdot a + pr_1(\text{read}(fl2)) \div \emptyset + \emptyset. \end{aligned}$$

In this example we observe that by symbolic evaluation the equation

$$W_P[I_{M+}(-,t)(b)] = W_P[I_{M-}(-,t)(b)]$$

is verified without explicitly referring to all reachable markings  $M$ . Repeating the same procedure for the inequality (B3) in

$$W_P[I_{M+}(-, t_3)(b)] = W_P[I_{M-}(-, t_3)(b)]$$

we obtain the *left-hand side*  $5 \cdot f$  and the *right-hand side*  $pr_2(\text{read}(f|2))$ , which is different. The reason for this defect is some lack of information in the left-hand side of (B3). As is known from the verification of Floyd-invariants, such difficulties are eliminated by the introduction of "auxiliary variables". These auxiliary variables are introduced for the verification process and do not affect the working of the program. Hence, they can be removed afterwards without influencing the validity of the correctness proof.

In the remainder of this section we will extend the net  $\mathcal{R}\mathcal{P}\mathcal{W}$  to nets  $\mathcal{R}\mathcal{P}\mathcal{W}_1$  and  $\mathcal{R}\mathcal{P}\mathcal{W}_2$  in such a way that theorem 4.6 can be applied to verify an equation (B3b) (also to be constructed). Then we will prove that (B3b) holds for  $\mathcal{R}\mathcal{P}\mathcal{W}_2$  if and only if (B3) holds for  $\mathcal{R}\mathcal{P}\mathcal{W}$ . Then all of the equations (B1), (B2) and (B3) will be verified by theorem 4.6.

To start with, we introduce a place "co\_read", which contains for every file  $f$  that is not in the place "write" but is read by  $0 \leq n \leq 5$  tasks (in the place "read"),  $5 - n$  instances of  $f$ . The auxiliary place allows to transform (B3) into an equality :

$$(B3a) \quad \forall M \in R(N) : pr_2(M(\text{read})) + M(\text{co\_read}) + 5 pr_2(M(\text{write})) = 5 F$$

The corresponding smCPN  $\mathcal{R}\mathcal{P}\mathcal{W}_1$  is shown in fig. 4.4. "co\_read" behaves like an ordinary complementary place, when "write" is empty :

$$\forall M \in R(N) : M(\text{write}) = \emptyset \Rightarrow pr_2(M(\text{read})) + M(\text{co\_read}) = 5 F$$

With an occurrence of  $t$ , however, all files  $y$  are removed from "co\_read". Since (B3a) implies (B3) it is sufficient to prove (B3a).

To use theorem 4.6, we introduce a place "deferred", where all instances of files  $f$  in "read" are collected while a task is writing on  $f$ . In the smCPN  $\mathcal{R}\mathcal{P}\mathcal{W}_2$  of fig 4.5 all instances of  $y$  in the places "read" and "co\_read" are moved to "deferred", when  $t_3$  is clearing them. The smCPN  $\mathcal{R}\mathcal{P}\mathcal{W}_2$  satisfies the following invariant equation :

$$(B3b) \quad \forall M \in R(N) : pr_2(M(\text{read})) + M(\text{co\_read}) + M(\text{deferred}) = 5 F$$

This is easily proved with theorem 4.6 by the corresponding incidence functions. (For the definition of deferred() see Def. 4.4.)

Our goal now is to deduce (B3a) for  $\mathcal{R}\mathcal{P}\mathcal{W}_1$ . To do this we use the fact that in

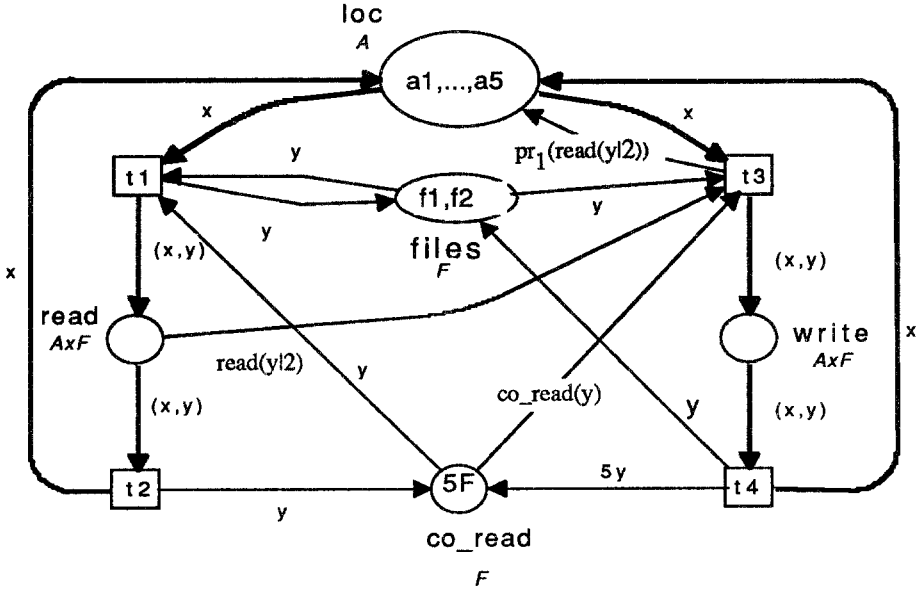


Fig. 4.4:  $\mathcal{Rpw}_1$ : readers and preemptive writers with place  $co\_read$

$\mathcal{Rpw}_2$  an occurrence of  $t_3$  in the colour  $(a, f)$  is clearing the places "read" and "co\_read" with respect to "f", hence after the occurrence of  $t_3$ :

$$(C1) \quad M(\text{read}(f_2)) = \emptyset \quad \text{and} \quad M(\text{co\_read}(f)) = \emptyset$$

This is holding up to the next occurrence of  $t_4$ , which is the only possibility to move token  $f$ . Therefore from (B3b) we conclude  $M[\text{deferred}(f)] = 5 \cdot f$  for such an interval, where we also have  $pr_2(M[\text{write}(f)]) = f$ , hence for each  $f \in F$ :

$$(C2) \quad M[\text{deferred}(f)] = 5 \cdot f \iff pr_2(M[\text{write}(f)]) = f \quad \text{or}$$

$$(C3) \quad M[\text{deferred}(f)] = 5 \cdot pr_2(M(\text{write}))$$

Since (C3) is true in  $\mathcal{Rpw}_2$  we can omit the place "deferred" and replace the term  $M(\text{deferred})$  in (B3b) by  $5 \cdot pr_2(M(\text{write}))$ . By this operation we have reconstructed the smCPN  $\mathcal{Rpw}_1$  and the invariant equation (B3a). As mentioned above this implies (B3) for  $\mathcal{Rpw}$ , which ends the proof of (B3).

## 5. Conclusion and Ongoing Research

The subject of this paper has its origin in a student's question. We believe the work allows a better understanding of how place-invariants of coloured nets relate to

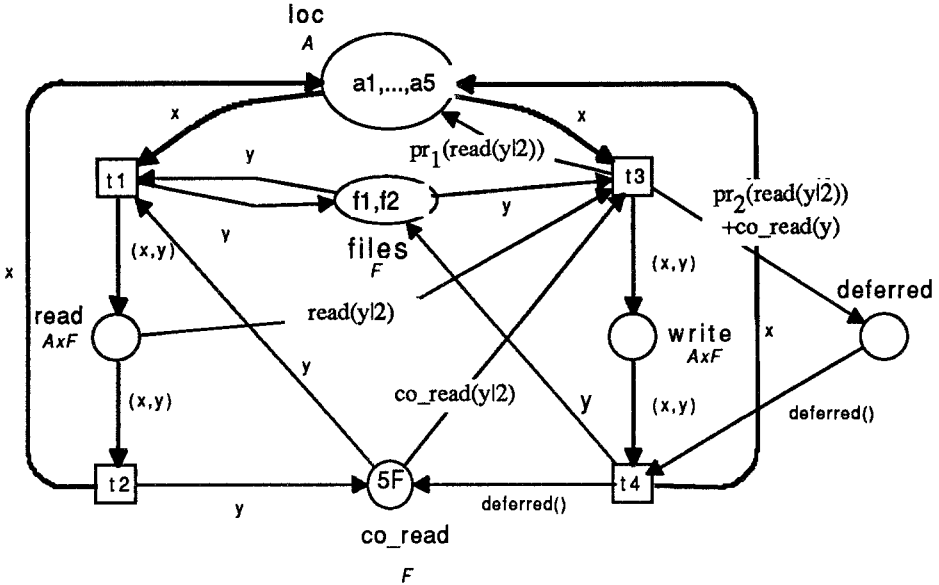


Fig. 4.5:  $Rpw_2$ : readers and preemptive writers with places "co\_read" and "deferred"

Floyd-invariants of ALGOL-like programs. Since the extended form of invariant equations, introduced in this paper, allows to relate internal values of tokens, we have done a step towards an invariance calculus for coloured nets with complex objects as tokens.

The extension of coloured nets, investigated in section 4, is very useful in many system modelling applications, such as preemptive scheduling and systems reset, but preserves the important property of place-invariant verification from the incidence matrices.

The main theorem of this paper (theorem 3.3) has been extended to hold for a broader class of equations. The extended theorem [Valk 93] proves weight functions, where non-linear parts may depend on more than one place. Hence also equations can be verified expressing non-linear relations between distinct variables. By this extension other types of applications are included, for instance a coloured net modelling a distributed algorithm for computing the greatest common divisor of a set of natural numbers.

## Acknowledgements

Part of this work has been done with Lab. MASI, Univ. Paris 6, in the common

project DAAD/PROCOPE, in particular together with Claude Dutheillet and Emmanuel Paviot-Adet. We also acknowledge valuable comments of several referees.

## 5. References

- [Floyd 67] Floyd, R. : Assigning meaning to programs. *Mathematical Aspects of Computer Science*, XIX American Mathematical Society (1967),19-32
- [Gries 81] Gries, D. :*The Science of Programming*, Springer, Berlin 1981
- [Hoare 69] Hoare, C.A.R. : An axiomatic approach to computer programming, *Comm. ACM* 12(1969), 576-580,583
- [Jensen 81] Jensen, K. : Coloured Petri Nets and the invariant method. *Theoretical Computer Science* 14(1981), 317-336
- [Jensen 87] Jensen, K. : Coloured Petri Nets, in *Petri Nets: Central Models and Their Properties*, in Brauer, W. et al. (Eds), *Lecture Notes in Computer Science* No 254, Springer, Berlin 1987, 248-299
- [Valk 78] Valk, R. :Selfmodifying Nets, a Natural Extension of Petri Nets, *Lecture Notes in Computer Science* No 62, Springer, Berlin 1978, 464-476
- [Valk 83] Valk, R. : Facts in Place/Transition Nets with Unrestricted Capacities, *Annales Univ. Scie. Budapestensis R. Eötvös, Sect.Comput. Tom. IV*, 1983, 97-105
- [Valk 93] Valk, R. : Extending S-invariants for Coloured and Selfmodifying Nets, *Techn. Report,Univ. Hamburg, Dep. of Computer Science*, 1993
- [Vautherin 85] Vautherin, J. : Non-linear invariants for safe coloured nets and application to the proof of parallel programs, *Proc. 6th European Workshop on Applications and Theory of Petri Nets*, Espoo, Finland, 1985