V. E. Kozura

UNFOLDINGS OF TIMED COLOURED PETRI NETS

Preprint
82

Novosibirsk 2000
In the paper [10] the unfolding technique was applied to coloured Petri nets (CPN) [8,9]. It was also shown in [10] how to use the unfolding technique taking into consideration symmetry or equivalence specifications. In [1] unfolding technique was applied to interval-timed Petri nets. This paper transfers this technique to interval-timed CPN and also considers the unfolding technique for timed CPN (TCPN) [8,9]. We require CPN to be finite, n-safe and containing only finite sets of colours.
Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

В.Е. Козюра

РАЗВЕРТКИ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ СО ВРЕМЕНЕМ

Препринт
82

Новосибирск 2000

© Институт систем информатики им А. П. Ершова СО РАН, 2000
1 INTRODUCTION

There are several techniques to avoid the “state explosion problem” in the state space analysis of Petri Nets. The stubborn set method, methods based on symbolic binary decision diagrams (BDD), methods using symmetry and equivalence properties of the state space and methods based on partial orders help us to avoid the problem in some cases[14].

McMillan in [12] has proposed an unfolding technique for Petri net analysis. In his works a finite prefix of maximal branching process, which is large enough to describe a system, has been considered instead of the reachability graph. The size of unfolding is exponential in the general case and there were few works which have improved in some way the unfolding definitions and the algorithms of unfolding construction [6, 11].

J. Esparsa has proposed a model-checking approach to unfolding of 1-safe systems analysis [5]. In [1] the model-checking technique using the net unfolding has been applied to timed Petri nets. In [7] LTL-based model-checking has been developed.

Unfolding of coloured Petri nets has been considered in the general case in [15] for using it in dependency analysis needed by the Stubborn Set method. In paper [10] the unfolding method, as it was developed in later works for ordinary Petri nets has been applied to coloured Petri nets (in the way they are described in [8,9]). It was also shown in [10] how to use the unfolding technique taking into consideration symmetry or equivalence specifications.

This paper transfers the unfolding technique from [1] to interval-timed CPN and also considers the unfolding generation for timed CPN as they are described in [8,9]. Let us notice here that the notion of unfolding with equivalence given in [10] is very useful when we want to obtain the complete unfolding of TCPN considered in [8,9].

The paper is organized as follows: chapter 2 gives the main definitions of the CPN’s theory and the subclass we are interested in, chapter 3 introduces the unfolding theory and gives the net example, chapter 4 describes the unfolding technique for interval-timed CPN and chapter 5 describes the unfolding technique for TCPN.
2 INTRODUCTION TO COLOURED PETRI NETS

In this section we briefly give the basic definitions related to CPN and de-
scribe the subclass of colours we will use in the paper. More detailed description
of coloured Petri nets can be found in [8,9].

Definition 2.1. A multi-set is a function m: S→N, where S is a usual set and N
is the set of natural numbers.

In the natural way we can define operations such as m1+m2, n·m, m1-m2, and
relations m1≤m2, m1<m2. Also |m| can be defined as |m|=\sum_{s\in S}m(s).

Let Var(expression) define the set of variables of expression and
Type(expression) define its type.

Definition 2.2. A coloured Petri net CPN is the net

\[ N = (S,P,T,A,N,C,G,E,I), \]

where S,P,T,A are the sets of colours, places, transitions, and arcs such that
P∩T = P∩A = T∩A=∅; N is a mapping N: A→P×T∪T×P; C is a colour func-
tion C: P→S; G is a guard function such that for all t∈T Type(G(t))=bool and
Type(Var(G(t))) ⊆ S; E is the function defined on arcs with Type(E(a)) = C(p)_{MS},
where p is the place from N(a) and Type(Var(E(a))) ⊆ S; and I is the initial
function defined on places such that for all p∈P Type(I(p)) = C(p)_{MS}.

A(t), Var(t), A(x,y), E(x,y) can be defined in the natural way.

Definition2.3. A binding b is a function from Var(t) such that b(v)∈Type(v) and
G(t)<b>. The set of bindings for t will be denoted by B(t)

Definition2.4. A token element is a pair (p,c) where p∈P and c∈C(p). The set of
all token elements is denoted by TE.

Definition2.5. A binding element is a pair (t,b) where t∈T and  b∈B(t). The set
of all binding elements is denoted by BE.

Definition2.6. A marking M is a multi-set over TE.

Definition2.7. A step Y is a multi-set over BE.

Definition2.8. A step Y is enabled in the marking M if for all p∈P
\[ \sum_{(t,b)\in Y}E(p,t)<b> \leq M(p) \]
and a new marking M1 is given by

\[ M_1(p) = M(p) - \sum_{(t,b)\in Y}E(p,t)<b> + \sum_{(t,b)\in Y}E(t,p)<b>. \]
Now we can define a subclass of colored Petri nets which is large enough to describe many interesting systems and still allows us to build a finite prefix of its branching process. In [10] more detailed description is given. The main idea is to consider only finite color domains $s \in S$. All functions defined in [8] and having the above described classes as their domains are allowed in our subclass. The same can be told about the variables, constants, operators and net expressions.

**Definition 2.9.** The CPN satisfying all the above-mentioned requirements is called $S$-finite.

**Definition 2.10.** The marking $M$ of a CPN is $n$-safe if $|M(p)| \leq n$ for all $p \in P$. A CPN is called $n$-safe if all of its reachable markings are $n$-safe. 1-safe net is also called safe.

**Definition 2.11.** A *preset* of an element $x \in P \cup T$ denoted by $\cdot x$ is the set
$$\cdot x = \{ y \in P \cup T \mid \exists a: N(a) = (y,x) \}.$$ 

A postset of $x$ denoted by $x^\cdot$ is the set
$$x^\cdot = \{ y \in P \cup T \mid \exists a: N(a) = (x,y) \}.$$ 

The CPN considered in this paper are the CPN satisfying three additional properties:

1. The number of places and transitions is finite.
2. The CPN is $n$-safe.
3. The CPN is $S$-finite.

If the opposite is not mentioned, the term CPN has the meaning of a CPN, satisfying these three properties.

### 3 UNFOLDINGS OF COLOURED PETRI NETS

Let $N$ be a Petri net. We will use the term *nodes* for both places and transitions.

**Definition 3.1.** The nodes $x_1$ and $x_2$ are in conflict, denoted by $x_1 \# x_2$, if there exist transitions $t_1$ and $t_2$ such that $\cdot t_1 \cap \cdot t_2 \neq \emptyset$ and $(t_1,x_1)$ and $(t_2,x_2)$ belong to the transitive closure of $N$ (which we denote by $R_i$). The node $x$ is in self-conflict if $x \# x$. We will write $x_1 \leq x_2$ if $(x_1,x_2) \in R_i$ and $x_1 < x_2$ if $x_1 \leq x_2$ and $x_1 \neq x_2$.

**Definition 3.2.** We say that $x$ co $y$, or $x \mid\mid y$, or $x$ concurrent $y$ if neither $x < y$ nor $x > y$ nor $x \# y$. 

7
Definition 3.3. An Occurrence Petri Net (OPN) is a usual Petri net $N = (P, T, N)$, where

(1) $P$ and $T$ are the sets of places and transitions,
(2) $N \subseteq P \times T \cup T \times P$ gives us the incidence function

satisfying the following properties:

(a) For all $p \in P \mid \bullet p \mid \leq 1$,
(b) $N$ is acyclic, i.e., the (irreflexive) transitive closure of $N$ is a partial order.
(c) $N$ is finitely preceded, i.e. for all $x \in P \cup T$ the set $\{ y \in P \cup T \mid y \leq x \}$ is finite which gives us the existence of Min($N$), the set of minimal elements of $N$ with respect to $R_t$ (which is considered to contain only the elements from $P$).
(d) no transition is in self conflict.

Every place $p \in P$ may have some tokens. The initial marking of an OPN $M_0$ of $N$ is defined by $M_0(p) = 1$ if $p \in \text{Min}(N)$ and empty otherwise. If for the transition $t \in T$ we have $M(p) > 0$ for all $p \in \bullet t$, then $t$ may occur and the obtained marking $M_1$ is given by $M_1 = M - M(\bullet t) + M(t \bullet)$.

Proposition 3.1. OPN is a 1-safe net.

Proof. The initial marking is 1-safe by definition. Using the restriction $\mid \bullet p \mid \leq 1$ from the OCPN definition, we have that, from the 1-safe marking by occurrence of any $t \in T$, we can obtain only 1-safe marking. Otherwise we have a contradiction either with the property (b) in the case $p \in \text{Min}(N)$ or with the above mentioned property (a) from the OCPN definition.

Definition 3.4. Let $N_1 = (P_1, T_1, N_1)$ and $N_2 = (P_2, T_2, N_2)$ be two Petri nets. A homomorphism $h$ from $N_2$ to $N_1$ is a mapping $h: P_2 \cup T_2 \rightarrow P_1 \cup T_1$ such that

(a) $h(P_2) \subseteq P_1$ and $h(T_2) \subseteq T_1$.
(b) for all $t \in T_2 h | \bullet = \bullet h(t)$.
   for all $t \in T_2 h | _t = t \rightarrow h(t)$.

Next we will give the definition from [10] of a branching process given for coloured Petri nets.

Definition 3.5: A branching process of a CPN $N_f = (S, P_1, T_1, A_1, N_1, C_1, G_1, E_1, I_1)$ is a tuple $(N_2, h, \varphi, \eta)$, where $N_2 = (P_2, T_2, N_2)$ is an OPN, $h$ is a homomorphism from $N_2$ to $N_f$, $\varphi$ and $\eta$ are functions from $P_2$ and $T_2$, respectively, such that
(a) $\phi(p) \in C(h(p))$.

(b) $\eta(t) \in B(h(t))$.

Other requirements are listed below:

(c) $\text{Min}(N_2) = M_0$.

Here and further the double equality operator means two equal multi-sets of token elements. This can also be written in the following way: for all $p_i \in P_1 \Sigma_{(p\in A)}\phi(p) = M_0(p_1)$, where $A = \{ p \in \text{Min}(N_2) \mid h(p) = p_1 \}$.

(d) $G(h(t)) \prec \eta(t) \succ$ for all $t \in T_2$.

(e) $\forall t' \in T_2 \mid (\exists a \in A_1: N_1(a) = (p,t) \text{ and } h(t') = t)$ \implies 
   
   $E(a) < \eta(t') = E(a) \prec \sum_{p' \in I} \phi(p')$, where $I = \{ p' \in \bullet t' \mid h(p') = p \}$.

(f) $\forall t' \in T_2 \mid (\exists a \in A_1: N_1(a) = (t,p) \text{ and } h(t') = t)$ \implies 
   
   $E(a) < \eta(t) = E(a) \prec \sum_{p' \in I} \phi(p')$, where $I = \{ p' \in \bullet (t,b) \mid h(p') = p \}$.

Important Note: Using the first two properties, we can associate a token element $(p,c)$ of $N_1$ with every place in $N_2$ and the binding element $(t,b)$ of $N_1$ with every transition in $N_2$. So we can further consider the net $N_2$ as containing the places which we identify with token elements of $N_1$, and transitions which we identify with binding elements of $N_1$. So we sometimes use them instead, like $h((t,b)) = t$ means $h(t') = t$ and $\eta(t') = b$ or $p \in \bullet (t,b)$ means $p \in \bullet t'$ and $h(t') = t$ and $\eta(t') = b$. Analogously, we can consider $(p,c) \in P_2$ as $p' \in P_2$ and $h(p') = p$ and $\phi(p) = c$. Also, $h(p,c) = p$ and $h(t,b) = t$.

It can be shown that any finite CPN has a maximal branching process (MBP) up to isomorphism (proposition 3.2). We can declare existence of the maximal branching process when considering the algorithm of its generation. As such an algorithm we choose the algorithm of unfolding generation proposed by McMillan [12] and applied to coloured Petri nets.

**Maximal Branching Process generation algorithm**

```
var: P_2,T_2,N_2;
// Places and transitions are natural numbers, N_2 is the set of pairs (m,n).
H_Table = {Ph_table[], Th_table[]} // This is a table for storing a homomorphism and functions \phi and \eta
// Ph: n \rightarrow (p,c), Th: m \rightarrow (t,b).
T_Fired; // The list of waiting binding elements.
m, n : integer; // The place and transition under construction.
```
Using H_Table for simplification of the algorithm, we sometimes write
//(p,c) and (t,b) instead of the corresponding n and m.

begin
H_Table:=empty;
N2= ( P2,T2,N2): = ∅; n:=1;m:=1;
for all p∈P1 such that |I(p)|>0 do
    for all c∈I(p) do
        begin
            add(n , P2);
            n:=n+1;
            GenTr({n-1});
        end;
end;
While (T_Fired ≠ ∅) do
    begin
        m0:= head(T_Fired) = (t,b);
        delete(m0,T_Fired);
        for all a∈A1 such that N1(a) = (t,p) do
            for all c∈E(a)<b> do
                begin
                    Ph_table[n]:=(p,c);
                    add((m0,n), N2);
                    add(n,P2);
                    n:=n+1;
                    GenTr({n-1})
                end;
        end;
return N2= ( P2,T2,N2);
end.

procedure GenTr(N);
begin
if (¬∃t∈T1 | N≤*t) then return
if Predecessors(N) has forward conflict then return
for all (t,b)∈TE such that h(N)=•t do
    if (t,b) is enabled in M==N then
        // i.e M = Ph_table[N]
        begin
            add( (N,m), N2);
        end;
end.
insert m = (t,b) in T_Fired in order of |LocalConfig(m)|
Th_table[m] := (t,b);
add(m,T2);
m := m + 1;
end;
for all n ∈ P2 \ N do
GenTr(N ∪ {n});
end.

**Proposition 3.2**[10]. The algorithm gives us the maximal branching process MBP(N_j) of N_j.

This branching process can be infinite even for the finite nets if they are not acyclic. We are interested to find a finite prefix of a branching process large enough to represent all the reachable markings of the initial CPN. This finite prefix will be called an unfolding of the initial CPN. In the next section we give the definitions of a configuration, cutoff points and the definition of unfolding of CPN.

**Definition 3.6.** A configuration C of an OPN N = (P,T,N) is a set of transitions satisfying the following two conditions:

1. t ∈ C ⇒ for all t_0 ≤ t : t_0 ∈ C
2. for all t_1,t_2 ∈ C : ¬(t_1#t_2).

**Definition 3.7.** A set X_0 ⊆ X of nodes is called a co-set, if for all t_1, t_2 ∈ X_0:
(t_1 co t_2).

**Definition 3.8.** A set X_0 ⊆ X of nodes is called a cut, if it is a maximal co-set with respect to the set inclusion.

Finite configurations and cuts are closely related. Let C be a finite configuration of an occurrence net, then Cut(C) = (Min(N) ∪ C) \ C is a cut.

**Definition 3.9.** Let N_j = (S_j,P_j,T_j,A_j,N_1,C_1,G_1, E_j,I_j) be a CPN and MBP(N_j) = (N_2, h, φ, η), where N_2 = (P_2,T_2,N_2), be its maximal branching process. Let C be a configuration of N_2. We define a marking Mark(C) == Cut(C) which is a marking of N_j. Operator “==” has the same meaning as in definition 3.5 Mark(C)(p) = Σ_{p' ∈ Cut(C)| h(p') = p} M_2(p').

**Definition 3.10.** Let N be an OPN. For all t ∈ T the configuration [t] = {t' ∈ T | t' ≤ t} is called a local configuration. (The fact that [t] is a configuration can be easily checked).
Let us consider the maximal branching process for a given CPN. It can be noticed that MBP(N) satisfies the completeness property, i.e., for every reachable marking M of N there exists a configuration C of MBP(N) (i.e., C is the configuration of OPN) such that Mark(C) = M. Otherwise we could add a necessary path and generate a larger branching process. This would be a contradiction with the maximality of MBP(N).

Now we are ready to define three types of cutoffs used in the definition of unfolding. The first two definitions can be found in [5] and [12]. The last is the definition given in [11].

**Definition 3.11.** A transition \( t \in T \) of an OPN is a **GT\( o \)-cutoff**, if there exists \( t_0 \in T \) such that \( \text{Mark}([t]) = \text{Mark}([t_0]) \) and \([ t_0] \subset [t]\).

**Definition 3.12.** A transition \( t \in T \) of an OPN is a **GT-cutoff**, if there exists \( t_0 \in T \) such that \( \text{Mark}([t]) = \text{Mark}([t_0]) \) and \( |[t_0]| < |[t]| \).

**Definition 3.13.** A transition \( t \in T \) of an OPN is a **EQ-cutoff**, if there exists \( t_0 \in T \) such that

1. \( \text{Mark}([t]) = \text{Mark}([t_0]) \)
2. \( |[t_0]| = |[t]| \)
3. \( \neg(t \parallel t_0) \)
4. there are no EQ-cutoffs among \( t' \) such that \( t' \parallel t_0 \) and \( |[t']| < |[t_0]| \).

**Definition 3.14.** For a coloured Petri net \( N \), an unfolding is obtained from the maximal branching process by removing all the transitions \( t' \), such that there exists a cutoff \( t \) and \( t < t' \), and all the places \( p \in t^* \).

If Cutoff = \( \text{GT}_{o}(\text{GT}) \)-cutoffs then the resulted unfolding is called \( \text{GT}_{o}(\text{GT}) \)-unfolding. \( \text{GT}_{o}(\text{GT}) \)-unfolding is also called a McMillan unfolding. If Cutoff = GT-cutoffs \( \cup \) EQ-cutoff then the resulted unfolding is called **EQ-unfolding**.

It has been shown that the McMillan unfoldings are inefficient in some cases. The resulting finite prefix grows exponentially, when the minimal finite prefix has only a linear growth. The following proposition can be formulated for these three types of unfoldings.

**Proposition 4.2 [10].** EQ-unfolding ≤ GT-unfolding ≤ \( \text{GT}_{o} \)-unfolding.

The following theorem gives correctness of the obtained unfoldings.

**Theorem 1 [10].** Let \( N \) be a CPN. Then for its unfoldings we have:

1. EQ-unfolding, GT-unfolding and \( \text{GT}_{o} \)-infolding are finite.
(2) EQ-unfolding, GT-unfolding and GT₀-infolding are safe, i.e., if C and C’ are configurations, then \( C \subseteq C' \Rightarrow \text{Mark}(C') \subseteq \text{Mark}(C) \).

(3) EQ-unfolding, GT-unfolding and GT₀-infolding are complete, i.e., \( M \in \text{Mark}(M_0) \Rightarrow \) there exists a configuration C such that \( \text{Mark}(C) = M \).

As an algorithm for unfolding generation we can use the algorithm of maximal branching process generation and add there the finitization function based on cutoff criteria. Exact algorithms can be found in [10].

As an example let us consider the CPN representing the problem of dining philosophers (Fig. 1). For this net we have \( \text{GT}_0 \)-unfolding = GT-unfolding = EQ-unfolding.

Unfoldings of this net are represented on Fig.2.
As it can be seen from the table below, the size of unfoldings is linear in the number of philosophers while the number of reachable markings is exponential.

<table>
<thead>
<tr>
<th>N</th>
<th>the unfolding sizes (the numbers of transitions)</th>
<th>Reachable Markings</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GT₀,GT,EQ-unfoldings 10</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>466</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2164</td>
</tr>
</tbody>
</table>

We measure the unfolding size by the number of transitions, because, when storing the information about each place in every reachable marking, we have the analogy with storing the fan-out places for every transition. (Anyway, the number of fan-out places is restricted by some constant and doesn’t spoil the linear growth of the unfolding size).
4 UNFOLDINGS OF INTERVAL-TIMED CPN

In this chapter we apply the technique of unfolding of timed PN from [1] to CPN with analogous time structure. Such CPN will be called interval-timed CPN. The nets considered in [1] are 1-safe and satisfy the Divergent-Time property (DT-nets). We require from CPN only to be finite, n-safe and S-finite. This means that 1-safety and Divergent-Time property are not necessary.

First, let us consider the Divergent-Time property. DT-property requires that if a place p loses its token at time t, then no token will arrive at p before time t+1. Although there are some classes of systems which can be modelled by the DT-nets, for many interesting systems DT-property isn’t true. For example, even for a simple model of communication protocols ABP described in [8,9], we have violation of DT-property. It is described in [1] how to avoid the requirement of holding DT-property for an unfolding construction. In this paper we choose another approach.

Definition 4.1. An interval-timed CPN (ITCPN) is a pair \( N_{IT} = (N, \chi) \), where \( N \) is a CPN and \( \chi \) is a transition inscription \( \chi : T \rightarrow \tau \subseteq \mathbb{N} \times \mathbb{N} \) (\( \tau \) consists of nonnegative integer intervals). For \( \chi(t) = (eft(t), lft(t)) \) we call \( eft(t) \) and \( lft(t) \) the earliest firing time and the latest firing time of \( t \), respectively.

Definition 4.2. A state of an interval-timed CPN \( N \) is a pair \( (M, I) \), where \( M \) is a marking of \( N \) and \( I \) is a clock vector \( I : T \rightarrow (\mathbb{N} \cup \{\$\}) \) such that either \( I(t) = \$ \) or \( I(t) < lft(t) \) for all \( t \in T \). The symbol \( \$ \) indicates that the corresponding transition is not enabled. A state is called consistent if for all \( t \in T \) \( I(t) \neq \$ \iff t \in \text{Enabled}(M) \). Only the consistent states will be considered in this paper. For an integer \( \theta > 0 \) and for all \( t \in T \), \( (I+\theta) \) is defined by \( (I+\theta)(t) = I(t)+\theta \) if \( t \in \text{Enabled}(M) \) and \$ otherwise.

Definition 4.3. The initial state \( (M_0, I_0) \) is defined by the initial marking \( M_0 \) and the initial clock vector \( I_0 \) such that \( I_0 = 0 \) if \( t \in \text{Enabled}(M_0) \) and \$ otherwise.

Definition 4.4. Two types of events are considered:

1. Tic-event: tic is fireable at the state \( (M, I) \) if for all \( t \in T \) \( I(t) < lft(t) \). In this case, the successor state \( (M_1, I_1) \) is given by \( M_1 = M \) and \( I_1 = (I+1) \). The tick-event is denoted by \( (M, I) \rightarrow ^{\text{tic}}(M_1, I_1) \).

2. Occur-event: An occur event is fireable at the state \( (M, I) \) if some transition \( t \) may occur with the binding element \( b \), i.e., if \( (t,b) \in \text{Enabled}(M) \) and \( eft(t) \leq I(t) \leq lft(t) \). In this case the successor state \( (M_1, I_1) \) is given by \( M_1(p) = M(p) - E(p,t)<b> + E(t,p)<b> \) and \( I_1(t’) \) is
\$t'$ \not\in \text{Enabled}(M_1),
0, \text{if } (t' = t) \text{ and } t' \in \text{Enabled}(M_1),
0, \text{if } (t' \neq t) \text{ and } t' \in \text{Enabled}(M_1) \text{ and } t' \not\in \text{Enabled}(M'), \text{ where}
M'(p) = M(p) - E(p,t)<b> \text{ and}
I(t') otherwise.

An occur event is denoted by \((M,I) \rightarrow^{(t,b)}(M_1,I_1)\).

Let us notice that the initial state is consistent and both occur- and tic-events preserve the consistency property.

Now we define the time expansion of an interval-timed CPN — \(X(\text{ITCPN})\) which captures the behaviour of the initial ITCPN and is an ordinary (untimed) CPN. As in the paper \([1]\) we will consider the part of the unfolding of \(X(\text{ITCPN})\) to be an unfolding of initial ITCPN (see below). In general, the size of \(X(\text{ITCPN})\) may be exponential in the size of the initial ITCPN, but the unfolding of ITCPN can be generated without constructing \(X(\text{ITCPN})\). However, we need the definition of time expansion of an interval-timed CPN to prove existence and the necessary properties of ITCPN’s unfolding. For any \(N_{IT} = (N,\chi)\) we, as usually, require \(N\) to be finite, n-safe and S-finite.

**Definition 4.5:** The time expansion of an interval-timed CPN \(N_{IT} = (N,\chi)\) is defined in the following way:

1. For every place \(p \in P\), a place \(p^c\) (complementary place) is introduced such that \(C(p) = C(p^c)\). The set of all complementary places is denoted by \(P^c\).
2. For each transition \(t \in T\), a new place \(p^t\) is introduced such that \(C(p^t) = \text{int with } l...lt(t)\), where the symbol $ is denoted by -1. The set of such places is denoted by \(P^t\)
3. The marking \(PL(I)\) is defined in the following way: \(PL(I)(p) = I(t)\) if \(p = p^t\) and empty otherwise. The state \((M,I)\) of the initial ITCPN is represented by the state \(X(M,I) = M \cup M^c \cup PL(I)\), where for all \(p^c \in P^c\): \(M^c(p^c) = nC(p) \setminus M(p)\) and \(n\) is the constant from the n-safety condition of the initial CPN.
4. For each marking \(M\) of ITCPN a new transition \(\text{tic}(M)\) is introduced such that the preset and postset of \(\text{tic}(M)\) are the set \(M \cup P^c \cup P^t\). (This means that \(\text{tic}(M) \cap P = \{p \mid M(p) \neq \text{empty}\}\). It is denoted by \(\star\text{tic}(M) \cap P = M\). The arc expressions are:
\[\forall p \quad (M(p) \neq \text{empty}): \quad N(a) = (p, \text{tic}(M,I)) \Rightarrow E(a) = M(p).
\[\forall p^c \in P^c \quad N(a) = (p^c, \text{tic}(M,I)) \Rightarrow E(a) = nC(p) \setminus M(p).
\[\forall p^t \in P^t \quad N(a) = (p^t, \text{tic}(M,I)) \Rightarrow E(a) = i, \quad \text{if } t \in \text{Enabled}(M), \text{ and empty otherwise.}
\[\forall p \quad (M(p) \neq \text{empty}): \quad N(a) = (\text{tic}(M,I), p) \Rightarrow E(a) = M(p).\]
∀ p^c ∈ P^c N(a) = (tic(M,I),p^c) ⇒ E(a) = n'C(p) \ M(p).
∀ p^t ∈ P^t N(a) = (p^t, tic(M,I)) ⇒ E(a) = i_t+1, if t ∈ Enabled(M), and empty otherwise.

guard[tic(M)] = ∃ i_t <lft(t).

Notice, that we consider only n-safe markings. The set of these transitions is denoted by Tic.

(5) For each marking M and each (t,b) ∈ BE we define a transition T\_{(t,b),M}.

The arcs are described below.

For all p ∈ P if ∃ a ∈ A_{IT} | N_{IT}(a) = (p,t) we define a_p,a_{pc} ∈ A such that

N(a_p) = (p, T_{(t,b),M}), N(a_{pc}) = (T_{(t,b),M}, p^c).
E(a_p) = E_{IT}(a)<b>, E(a_{pc}) = E_{IT}(a)<b>.

For all p ∈ P if ∃ a ∈ A_{IT} | N_{IT}(a) = (t,p) we define a_p,a_{pc} ∈ A such that

N(a_p) = (T_{(t,b),M}, p), N(a_{pc}) = (p^c, T_{(t,b),M}).
E(a_p) = E_{IT}(a)<b>, E(a_{pc}) = E_{IT}(a)<b>.

For all t' ∈ T_{IT} we define a_{1,t'},a_{2,t'} ∈ A such that

N(a_{1,t'}) = (p^t, T_{(t,b),M}), N(a_{2,t'}) = (T_{(t,b),M}, p^t),
E(a_{1,t'}) = i_{t'},
E(a_{2,t'}) = -1 if t' ∉ Enabled(M_1),
0 if (t' = t) and t' ∈ Enabled(M_1),
0 if (t' ≠ t) and t' ∈ Enabled(M_1) and t' ∉ Enabled(M'),
where M'(p) = M(p) - E(p,t)<b> + E(t,p)<b>,
i_{1,t'} — otherwise.

The set of these transitions is denoted by Fire.

The whole CPN constructed is

N_{X|ITCPN} = (S_X,P_X,T_X,A_X,N_X,C_X,G_X,E_X,I_X), where

S_X = S_{IT} ⊃ C(P^t),
P_X = P ⊃ P^c ⊃ P^t,
T_X = Tic ⊃ Fire,
C_X(p) = C_X(p^c) = C(p),
C_X(p^t) = int with -1..lft(t),
the sets A_X,N_X,G_X,E_X are described in the definition,
the initial marking M_{X_0} = M_0 ∪ M_0^c ∪ PL(I_0).

Now let us write some comments to each of these six points.
As shown in [1], we have some problems when modelling the clock events during the time expansion construction. First, if we introduce a tic transition for each state \((M,I)\) when tic is possible, we can come to a situation when, instead of this tic transition, the tic transition for \((M',I')\) fires where \(M' \subset M\). This is the reason for introducing the complementary places.

For every transition \(t\) we introduce the place \(p^t\), where the clock position for \(t\) will be stored.

In this point we define a marking \(X(M,I)\) of time expansion using complementary and clock places.

These transitions model time-events. The arc expressions in the definition could be written using the variables evaluations of which could be moved to the guard functions. Such a definition would be more in the style of CPN description in [8,9]. However, we leave the arc functions as they are to make the definition more observable. Let us notice that we also could make the set of tic transitions based on the subsets \(T' \subset T\) \((\text{tic}(T')\) instead of basing them on the set of markings. In this case the descriptions of \(M\) will be transferred to the guard functions. The variant presented in the definition is chosen to retain the analogy with the article [1].

These transitions model the occur-events and additionally update clock vectors. As shown in [1] the clock updating is needed to model firing of transitions. Since we represent the clock by the unique place for each transition, we don't need to have the set of transitions parameterized by the clock positions.

As it was written earlier, we don't require our CPN to satisfy DT-property. This means that we have to store in some way the "intermediate" markings. This is needed when some place \(p\) loses its token at the time \(t\) and at the same time some token arrives at \(p\). We elaborate such an "intermediate" marking in \(E(a_{1,t'})\) and \(E(a_{2,t'})\).

From the definition we make a conclusion about existence of \(X(\text{ITCPN})\)’s unfolding. Since the time expansion is finite, n-safe and S-finite, we obtain, accordingly to theorem 1, finiteness, safety and completeness of the generated unfolding.

We can also consider the part of the unfolding of \(X(\text{ITCPN})\) to be an unfolding of initial ITCPN (see below). The adequacy of this approach is given by the theorem below. Let us first give the following definition.
**Definition 4.6:** A marking $M$ of $X$(ITCPN) is called consistent iff

1. $|M(p^t)| = 1$ for all $t \in T_T$.
2. $M(p) \cup M(p^c) = n^*C(p)$.
3. $M(p^t) = -1 \iff t \not\in \text{Enabled}(M \cap P)$ for all $t \in T$.

Let us notice that the initial state is consistent and by the definition of time expansion any state reached from the consistent state is consistent.

**Theorem 2.** Let $N_{IT} = (N,\chi)$ be an ITCPN and we constructed its time expansion $N_X$. Then we have the following results:

1. A tic-event can occur at $(M,I)$ and $(M,I) \to^{\text{tic}}(M,I') \iff \text{tic} \in \text{Tic}$ is enabled in $M \cup M^c \cup \text{PL}(I)$ and $M \cup M^c \cup \text{PL}(I) \to^{\text{tic}} M \cup M^c \cup \text{PL}(I')$.
2. $(t,b)$ can occur at $(M,I)$ and $(M,I) \to^{(t,b)}(M',I') \iff T \in \text{Fire}$ is enabled in $M \cup M^c \cup \text{PL}(I)$ and $M \cup M^c \cup \text{PL}(I) \to^T M' \cup M'^c \cup \text{PL}(I')$.
3. The (consistent) state $(M,I)$ is reachable in $N_{IT} \iff$ the (consistent) state $M \cup M^c \cup \text{PL}(I)$ is reachable in $N_X$. In particular, $M$ is reachable in $N_{IT} \iff M = M' \cap P$ for some reachable marking $M'$ of $N_{IT}$.

**Proof:**

1. $(\Rightarrow)$ Let tic be possible in the state $(M,I)$. There exists tic$(M) \in T_X$ and, by the definition of time expansion,

$$E_X(a) = M(p) \text{ if } N_X(a) = (p,\text{tic}(M)) \text{ or } N_X(a) = (\text{tic}(M),p).$$

In the case of $P^c$ we get the whole set $P^c$ as a pre and post set for tic$(M)$ and the markings on $P^c$ remain unchanged.

From definitions of $E(a^i)$ we can conclude that $\forall t \in T_X$ if $t \in \text{Enabled}(M)$ then tic$(M)$, for $t \in \text{Enabled}(M)$, increases the value of the token in the place $p^t$ by one and leaves it untouched otherwise.

So, from the definition of time expansion, we obtain that tic$(M)$ starts from the marking $M \cup M^c \cup \text{PL}(I)$ and, after, occurring leaves the marking $M \cup M^c \cup \text{PL}(I')$.

$(\Leftarrow)$ If tic has occurred in $M \cup M^c \cup \text{PL}(I)$, then it is a tic$(M)$ transition (because of marking control made by complementary places).

Since tic$(M)$ has occurred, it satisfies the guard function:

$$\forall i^t | i^t < \text{lft}(t) \Rightarrow \text{PL}(I)(p) = I(t) < \text{lft}(t).$$

This means that tic is enabled at the state $(M,I)$ of the initial net $N$. Accordingly to $E(a^i)$ definition, we have that clock places $\text{PL}(I')$ correspond to the clock
function obtained by occurrence of tic at the state \((M,I)\). This means that \((M,I) \rightarrow^{\text{tic}}(M,I')\).

(2) 
\(\Rightarrow\) If \((t,b)\) is enabled in the state \((M,I)\), then \(T_{M,(t,b)} \in T_X\).

As in the case of tic transitions, from the definition of time expansion we obtain: 
\[ M \cup M^c \cup PL(I) \rightarrow^{TM,(t,b)} M' \cup M'^c \cup PL(I'). \]

For all \(t'\) the expression on the arc \(a_{1,t'}\) contains the input variables \(i^{t'}\), and the expression on the output arc \(a_{2,t'}\) is constructed accordingly to the occur-event definition.

(3) 
\(\Leftarrow\) If \(T\) has occurred in \(M\), then (because of marking control made by complementary places) \(T = T_{M,(t,b)}\) for some \((t,b)\in BE\). This means that \((t,b)\) is enabled in the state \((M,I)\) and, after occurring (by the definition of time expansion), gives us the marking \((M',I')\).

(3) 
\(\Rightarrow\) Let \((M_0,I_0) \rightarrow^{\sigma}(M,I)\). In the proof we will use induction on the length of \(\sigma\). If |\(\sigma\)| = 0, then by definition \(M_0 \cup M_0^c \cup PL(I_0)\) is the initial state and therefore is reachable.

If \(\sigma = \sigma 1 + \{ t \}\), \((M_0,I_0) \rightarrow^{\sigma 1}(M',I') \rightarrow^{t}(M,I)\) and \(M' \cup M'^c \cup PL(I')\) is reachable in \(N_X\), then we have two cases:

(a) \(t = \text{tic}\). Using point (1) of this theorem we can conclude about the reachability of \(M \cup M^c \cup PL(I)\).

(b) \(t = (t,b)\). Using point (2) of this theorem we conclude about the reachability of \(M \cup M^c \cup PL(I)\).

\(\Leftarrow\) Let \(M_0 \cup M_0^c \cup PL(I_0) \rightarrow^{\sigma} M \cup M^c \cup PL(I)\). In points (1) and (2) we have the equivalence properties. So, as in the previous part, we can apply induction on |\(\sigma\)|.

If |\(\sigma\)| = 0, then \(M_0 \cup M_0^c \cup PL(I_0)\) is initial and by the definition of time expansion we obtain that \((M_0,I_0)\) is initial in \(N\) and therefore is reachable.

If |\(\sigma\)| \(\neq\) 0, then we can consider two cases as in the previous part and conclude, using properties (1) and (2) of this theorem, that the state \((M,I)\) is reachable.

As mentioned earlier, the time expansion of CPN is used only to prove existence of a finite, safe and complete unfolding of ITCPN. Below we give the definition of a reduced unfolding which is obtained from the unfolding of \(X(\text{ITCPN})\) by removing the parts with unnecessary information and can be constructed directly from the ITCPN. We consider the reduced unfolding of \(X(\text{ITCPN})\) to be an unfolding of initial ITCPN. Although the exact algorithm description is out of
the scope of the paper, the basic idea of how to construct a reduced unfolding directly from ITCPN will be given.

**Definition 4.7.** Let \( N \) be an ITCPN and the finite unfolding \( \text{Unf}(N_X) \) of its time expansion be constructed, then a *reduced unfolding* is obtained from \( \text{Unf}(N_X) \) in the next two steps:

1. Remove all the places \( p^c \) and \( p^t \) from the unfolding and all the incidental arcs.
2. Add the names \((t, b)\) and tic to \( T_{(t,b),M} \) and tic(M) respectively.

The reduced unfolding is denoted by \( R(\text{Unf}(N_X)) \).

The configuration \( C = (t_1...t_n) \) of \( \text{Unf}(N_X) \) has a corresponding configuration \( C' = (t'_1...t'_n) \) of \( R(\text{Unf}(N_X)) \) such that if \( t_i = T_{M,(t,b)} \) then \( t'_i = (t,b) \) and if \( t_i = \text{tic}(M) \) then \( t'_i = \text{tic} \) and vice versa.

It also follows from the reduced unfolding definition that

\[
\text{Mark}_{\text{Unf}(N_X)}(C) \cap P = \text{Mark}_{R(\text{Unf}(N_X))}(C').
\]

Let us notice here that it would be more in the style of [8,9] to consider steps Y instead of single transitions in the theorem and definitions. However, the approach chosen here gives us a simple and clear way of how to describe the unfoldings of ITCPN. Below we give the net example and its GT-unfolding.

![Fig. 3 The Dining Philosophers](image-url)
Now we give the basic idea of how to modify the unfolding algorithm for CPN into the algorithm of reduced unfolding generation for a given interval-timed CPN. At each step, considering a co-set of places to be a preset for a given binding element \((t,b)\), we should calculate the function \(I\) for all \(t\) such that \(t < P'\), where \(P'\) are the places from the current co-set. This takes \(O(N_T N_P)\) time. So we should multiply the complexity of the unfolding algorithm by \(N_T N_P\). To find a deadlock by the method described in [10,13], we should consider cutoffs occurring after the tic transition when for all \(t \ I(t) = S\) as a decision of the system of equations.

**5 UNFOLDINGS OF TIMED CPN**

In this section we describe the unfolding technique for timed CPN (TCPN) as they are represented in [8,9]. There a more detailed description of timed CPN can be found. Here we briefly give the main definitions.

**Definition 5.1.** A *timed multi-set* \(tm\), over a non-empty set \(S\), is a function \(tm \in [S \times \mathbb{N} \rightarrow \mathbb{N}]\) such that the sum \(tm(s) = \sum_{n \in \mathbb{N}} tm(s,n)\) is finite for all \(s \in S\) (we
consider time values to be a set of integers. \( \text{tm}(s) \) is the number of appearances
of \( s \). The list \( \text{tm}[s] = [n_1, n_2, ..., n_{\text{tm}(s)}] \) is defined as containing the time values for
which \( \text{tm}(s,n) \neq \emptyset \). The timed multi-set is represented by a formal sum
\( \sum_{(s \in S)} \text{tm}(s)'s@\text{tm}[s] \). \( \text{tm}(s) \) is called a coefficient of \( s \). For a timed multi-set \( \text{tm} \),
an ordinary multi-set \( \text{tm}_U \) is defined by \( \sum_{(s \in S)} \text{tm}(s)'s \).

Analogously, for an ordinary multi-set \( m \) and a time value \( n \) we define
\( m_n = \sum_{(s \in S)} m(s)'s@\text{tm}[n,n...n] \).

As an example we consider the multi-set
\( \text{tm} = 2'(q,3)@[11,26] + 1'(q,4)@[526] \) for which we have \( \text{tm}[(q,3)] = [11,26] \),
and \( \text{tm}_U = 2'(q,3) + 1'(q,4) \).

For a timed multi-set \( \text{tm} \) and a time value \( n \), the multi-set \( \text{tm}_n \) is defined as
\( \text{tm}_n = \sum \text{tm}(s)'s@\text{tm}[s]_n \), where \( \text{tm}[s]_n \) is the list obtained from \( \text{tm}[s] \) by adding \( n \)
to each time value.

Let \( a = [a_1, a_2, ..., a_m] \) and \( b = [b_1, b_2, ..., b_n] \) be two lists over the set of natural num-
bers. \( a \leq b \) iff \( m \leq n \) and \( a_i \geq b_i \) for all \( i=1..m \).

When \( a \leq b \), \( b-a \) is defined as a list of length \( n-m \) which is obtained from \( b \) in
the following way. We remove from \( b \) the largest element which is smaller than
\( a_1 \). From the remaining list, we remove the largest time value which is smaller
than \( a_2 \), etc.

**Definition 5.2.** _Comparison_ between timed multi-sets is defined in the following
way, for all timed multi-sets \( \text{tm}_1, \text{tm}_2 \):

1. \( \text{tm}_1 \leq \text{tm}_2 \Leftrightarrow \forall s \in S: \text{tm}_1[s] \leq \text{tm}_2[s] \).

When \( \text{tm}_1 \leq \text{tm}_2 \) we also define the subtraction:

2. \( \text{tm}_2 - \text{tm}_1 = \sum_{(s \in S)} (\text{tm}_2(s) - \text{tm}_1(s))'s@((\text{tm}_2[s] - \text{tm}_1[s]) \).

**Definition 5.3.** A _timed CPN_ is a pair \( \text{TCPN} = (\text{CPN}, t_0) \) such that

1. CPN satisfies the requirements of an ordinary CPN, and the types of \( E(a) \)
   and \( I(p) \) are allowed to be timed or untimed multi-sets.
2. \( t_0 \) is a natural number called the start time.

Timed CPN often contain one or more colour sets \( S \) which are untimed. This
means that the token of type \( S \) are required to be always available, independently
of any time constraints.

**Definition 5.4.** A _marking_ is a timed multi-set over \( TE \). The initial marking \( M_0 \) is
given by \( M_0(p) = I(p)_{t_0} \). A state is a pair \( (M,t) \) and the initial state is a pair
\( (M_0,t_0) \).
Definition 5.5. A step $Y$ is enabled in a state $(M_1,t_1)$ at time $t_2$ iff the following properties are satisfied:

1. for all $p \in P$: $\sum_{(t,b) \in Y} E(p,t) < b > t_2 \leq M_1(p)$.
2. $t_1 \leq t_2$.
3. $t_2$ is the smallest time value for which there exists a step satisfying (1) and (2).

When such a step occurs, we obtain a new marking $M_2$ given by

$$M_2(p) = M_1(p) - \sum_{(t,b) \in Y} E(p,t) < b > t_2 + \sum_{(t,b) \in Y} E(t,p) < b > t_2.$$

It is more natural to use single transitions instead of steps for the unfolding generation. So, we will define, as in the previous part, two kinds of transitions.

Definition 5.6. Two kinds of events are defined in the following way:

(a) If in a state $(M,r)$ the step $Y = \{(t,b)\}$ is enabled at time $r$, then the occur-event is enabled in the state $(M,r)$ and the obtained state $(M',r)$ is defined as a result of occurring $Y$. We denote it by $(M,r) \to^{(t,b)} (M',r)$.

(b) If in a state $(M,r)$ some step $Y$ is enabled at the time $r' > r$, then the tic-event is enabled in the state $(M,r)$. We denote it by $(M,r) \to^{\text{tic}} (M',r+1)$.

We will consider only occur- or tic-events when considering the TCPN evaluation. As it was made for ITCPN, below we give the definition of time expansion and reduced unfolding of TCPN.

Definition 5.7. A time expansion of TCPN $N = (S,P,T,A,N,C,G,E,I)$ denoted by $X(TCPN)$ is the coloured Petri net described below.

1. The set of places is $P \cup P_c \cup p^i$, where
   
   $C_X(p) = C(p)$ if $C(p)$ is an untimed multi-set,  
   
   $C_X(p) = C(p) \times \text{int}$ otherwise.  
   
   $C_X(p^c) = C(p)$ for all $p \in P$,  
   
   $C_X(p^i) = \text{int}.$

   We define the unique clock place for the whole net and consider the time stamps as parts of the colour description.

2. The marking $X(M,r)$ is defined in the following way.
   
   $X(M,r)(p) = \sum_{s \in M(p) \cup} (s \times \text{tm}[s])$, if $C(p)$ is a timed multi-set (further we will write such a sum as $M(p) \cup \otimes \text{tm}[M(p)]$). We suppose $M(p) \cup$ to be sorted in some way which gathers the same color elements in one part of a list and $\text{tm}[M(p)]$ to be a respective list of time values, where for all $c \in M(p)$ the parts $\text{tm}[c]$ are sorted in the ascending order;
\[
X(M,r)(p) = \begin{cases} 
M(p) & \text{otherwise;} \\
 \n' \cap (p) \setminus M(p) & , \\
X(M,r)(p^c) = r. 
\end{cases}
\]

(3) For all markings \(M\) such that some step \(Y = \{(t,b)\}\) is colour-enabled in \(M_U\), we define a transition \(\text{tic}(M)\) which has \(M \cup P^c \cup p^1\) as its pre- and postsets (also as in the previous chapter: \(\text{\text{tic}(M)} \cap \mathcal{P} = \{p \mid M(p) \neq \text{empty}\}\)).

\[
N_X(a) = (p, \text{tic}) \text{ or } N_X(a) = (\text{tic}, p) \implies 
E_X(a) = M(p) \otimes J_{M(p)}, \text{ if } C(p) \text{ is a timed multi-set (} J_{M(p)} = \{x_1, \ldots, x_{\mid M(p)\mid}\} \text{ is the set of integer variables, } M(p)_U \text{ is sorted as in the previous point);} \\
E_X(a) = M(p) \text{ otherwise,} \\
N_X(a) = (p^c, \text{tic}) \text{ or } N_X(a) = (\text{tic}, p^c) \implies E(a) = \n' \cap (p) \setminus M(p) , \\
N_X(a) = (p^1, \text{tic}) \implies E_X(a) = x^l , \\
N_X(a) = (\text{tic}, p^1) \implies E_X(a) = x^l + 1.
\]

Let for all \(c \in E(a)_U, [r_{c1}, \ldots, r_{cm}]\) denote the ascending list of time values and for all \(c \in M(p)_U, [x_{c1}, \ldots, x_{cm}]\) denote the respective sublist of integer variables.

The function \(\text{guard(\text{tic}(M))}\) is defined in the following way.

First, the sublists \([x_{c1}, \ldots, x_{cm}]\) are sorted for all \(c \in M(p)_U: x_{c1} \leq x_{c2} \leq \ldots \leq x_{cm}\). Then, \(\forall (t,b) \in \text{color_enabled}(M_U) \exists a \text{ such that } N(a) = (p, t)\) and at least one \(c \in E(a)_U\) satisfies the condition: \(\forall j = 1, \ldots, cm \mid x_{cj} \geq r_{cj} + x^l\). The list of \(x_{cj}\) is longer than that of \(r_{cj}\), while \((t, b)\) is color-enabled in \(M_U\). Formally this can be written using the propositional logic operations.

The set of tic transitions is denoted by \(\text{Tic}\).

(4) \(\forall (t, b) \in \text{BE} \forall M_U\) we define a transition \(T_{M,(t, b)} \in T_X\) with the arc expressions described below.

If \(N(a) = (p, t)\) and \(C(p)\) is a timed multi-set, we define \(a^p, a^p, a^p, a^p\) and \(a^c\) such that

\[
N_X(a^p, out) = (p, T_{M,(t, b)}), \\
N_X(a^p, in) = (T_{M,(t, b)}, p), \\
N_X(a^c) = (T_{M,(t, b)}, p^c), \\
E(a^p, out) = M(p) \otimes L_M, \text{ where } L_M = \{l_{11}, \ldots, l_{\mid M(p)\mid}\}, \\
E(a^p, in) = (M(p)_U \otimes L_M) - (E(a)_U \otimes L_a), \text{ where } L_a = \{l_{a1}, \ldots, l_{a_{\mid E(a)\mid}}\} \text{ is the set of integer variables} \\
\quad (M(p)_U \text{ is sorted in the standard way}), \\
E(a^c) = E(a)_U.
\]

If \(C(p)\) is an untimed multi-set, then we define \(a^p\) and \(a^c\) such that

\[
N_X(a^p) = (p, T_{M,(t, b)}), \\
N_X(a^c) = (T_{M,(t, b)}, p^c).
\]
If \( N(a) = (t, p) \), we define \( a^p \) and \( a^c \) such that
\[
N_X(a^p) = (T_{M,t,b}, p),
N_X(a^c) = (p^c, T_{M,t,b}),
E_X(a^p) = E(a) \otimes Y_a, \text{ where } Y_a = \{y_1 \ldots y_{|E(a)|}\} \text{ is the set of integer variables}
\]
if \( C(p) \) is a timed multi-set,
\[
E_X(a^p) = E(a)^{<b>}, \text{ otherwise,}
E_X(a^c) = E(a)^{<b>}.
\]

For the sets \( L_a, Y_a \) we define the sets \( L_a', Y_a' \) of the corresponding time coefficients of \( E(a) \) sorted in the standard way (the same color-elements are in one part of the list and the respective lists of time values \( tm[c] \) are sorted in the ascending order).

We also define the arcs \( a^{t,1} \) and \( a^{t,2} \) such that
\[
N_X(a^{t,1}) = (p^t, T_{M,t,b}),
N_X(a^{t,2}) = (T_{M,t,b}, p^t),
E_X(a^{t,1}) = E_X(a^{t,2}) = x^t.
\]

The function guard\((T_{M,t,b})\) is defined in the following way:
\[
\forall a \mid N(a) = (p, t) \forall c \in E(a)^{<b>} \cup I_a, l_{a,1} \leq l_{a,2} \leq \ldots \leq l_{a,m},
\forall a \mid N(a) = (t, p) \forall c \in E(a)^{<b>} \cup I_{a,1} \leq y_{a,2} \leq \ldots \leq y_{a,k},
\forall a \mid N(a) = (p, t) \forall i = 1 \ldots |E(a)| l_{a,i} \leq x^i + l_{i,1}', \text{ where } l_{a,i} \in L_a, l_{i,1}' \in L_{a,1}',
\forall a \mid N(a) = (t, p) \forall i = 1 \ldots |E(a)| y_i = x^i + m_{i,1}', \text{ where } y_i \in Y_a, y_{i,1}' \in Y_{a,1}',
\forall a \mid N(a) = (p, t) \forall i = 1 \ldots |E(a)| \forall j \text{ such that } M(p)_{Uj} = E(a)^{<b>} \cup I_i : (l_j \leq x^i + l_{i,1}') \Rightarrow (l_j \leq l_{a,i}).
\]

Let us remind that we keep \( M(p) \) sorted.

The set of such transitions is denoted by Fire.

The whole CPN so constructed is \( N_X(TCPN) = (S_X, P_X, T_X, A_X, N_X, C_X, G_X, E_X, I_X) \), where
\( S_X \) is defined in the description of the function \( C \),
\( P_X = P \cup P^c \cup p^t \),
\( T_X = Tic \cup Fire \),
\( C_X(p^c) = C(p)_{U} \),
\( C_X(p) = C(p)_{U} \times \text{ int if } C(p) \text{ is the timed multi-set and} \)
\( C_X(p) = C(p) \text{ otherwise,} \)
\( C_X(p^t) = \text{ int,} \)
the sets \( A_X, N_X, G_X, E_X \) are described in the definition,
the initial marking \( M_{X_0} = X(M_0, t_0) \).
While the unique time counter doesn’t make such problems with the clock updating as an individual timer for every transition, we don’t need any updating. Notice that we don’t use the complementary places in the time elaboration.

While there are no constraints on the time value r for a given marking M, every n-safe state (M,r) can be called consistent. In the time expansion we define the consistent marking in the following way:

**Definition 5.8.** A marking M of X(TCPN) is called consistent iff

1. |M(p')| = 1.
2. M(p) ∪ M(p^c) = n' C(p).

Let us notice that the initial state is consistent and by the definition of time expansion any state reached from the consistent state is consistent. Every reachable marking is consistent and therefore has the type of X(M,r) for some consistent state (M,r). As in the previous part we can prove the theorem that gives us the relationship between the TCPN and its time expansion.

**Theorem 3.** Let us have TCPN N_T=(CPN, t_0) and its time expansion N_X be constructed. Then we have the following results.

1. A tic-event can occur at (M,r) and (M,r) →^{tic} (M, r+1) ⇔ tic ∈ Tic is enabled in X(M,r) and X(M,r) →^{tic} X(M,r+1).
2. (t,b) can occur at (M,r) and (M,r) →^{(t,b)} (M',r) ⇔ T ∈ Fire is enabled in X(M,r) and X(M,r) →^{T} X(M',r).
3. The (consistent) state (M,r) is reachable in N_T ⇔ the (consistent) marking X(M,r) is reachable in N_X. In particular, M is reachable in N_T ⇔ M = M' ∩ P for some reachable marking M' of N_X.

**Proof:**

1. (1) ⇒) Let tic be possible in the state (M,r). A transition tic(M) exists by the definition of time expansion (because some step Y is colour-enabled in M_0).

   By the definition of time expansion (3), tic (M) has M ⊇ P^c ∪ p^i as its pre- and post-set. By the definition of E_X(a), tic(M) takes the marking X(M,r) and after occurring leaves the marking X(M,r+1).

   ⇐) If tic has occurred in X(M,r), then it is a tic(M) transition (because of marking control made by the complementary places). tic(M) satisfies the guard function, which means that if some step Y is colour-enabled in M, then it is not time-enabled (some step Y is enabled by the time expansion definition). Therefore, the tic event is enabled in (M,r) and it changes (M,r) into (M,r+1).
(2) \[ \Rightarrow \] If \((t,b)\) is enabled in \((M,r)\), then \(T_{M,(t,b)} \in TX\). For a timed place this transition takes the whole marking \(M\) and, accordingly to the guard function, chooses the elements from \(E(a)<b>\) satisfying the time conditions described above in the TCPN definitions (i.e., it takes the token with the maximal time value \(l_{a,i} \leq x^t + l_i\)). For the untimed place \(p\), \(T_{M,(t,b)}\) takes \(E(a)<b>\) from \(p\), where \(N(a) = (p,t)\).

After occurring, \(T_{M,(t,b)}\) brings tokens to the postset of \(t\) accordingly to the TCPN definitions (\(E(a)<b>\) with the time values \(y_i = x^t + m_i^*\) in the case of time places). Also \(T_{M,(t,b)}\) returns back unused tokens from the preset.

Therefore, we have \(X(M,r) \rightarrow^{TM,(t,b)} X(M',r)\), where \(M'\) is obtained from \(M\) after occurrence of \((t,b)\).

\[ \Leftarrow \] If \(T\) has occurred in \((M,r)\), then it is \(T_{M,(t,b)}\) for some \((t,b)\) (because of marking control made by the complementary places). This means that \((t,b)\) is enabled in the state \((M,r)\) and by the definition of time expansion (using the considerations of the previous point) gives us the marking \((M',r)\).

(3) \[ \Rightarrow \] Let \((M_0,I_0) \rightarrow^\sigma (M,I)\). In the proof we will use induction on the length of \(\sigma\).

If \(|\sigma| = 0\), then by definition \(X(M_0,t_0)\) is the initial state and therefore is reachable.

If \(\sigma = \sigma_1 + \{ t \} \), \((M_0,t_0) \rightarrow^{\sigma_1} (M',t') \rightarrow^t (M,t)\) and \(X(M',t')\) is reachable in \(NX\), then we have two cases:

(a) \(t = tic\). Using point (1) of this theorem, we can conclude about the reachability of \(X(M,t)\).

(b) \(t = (t,b)\). We conclude about the reachability of \(X(M,t)\) using point (2) of this theorem.

\[ \Leftarrow \] Let \(X(M_0,t_0) \rightarrow^\sigma X(M,t)\). In points (1) and (2) we have the equivalence properties. So, as in the previous part, we can apply induction on \(|\sigma|\).

If \(|\sigma| = 0\), then \(X(M_0,t_0)\) is initial and by the definition of time expansion we obtain that \((M_0,t_0)\) is initial in \(N\) and therefore is reachable.

If \(|\sigma| \neq 0\), then we can consider two cases as in the previous part and conclude, using properties (1) and (2) of this theorem, that the state \((M,t)\) is reachable.

Definition 5.8. Let \(N\) be a TCPN and the finite unfolding \(Unf(N_X)\) of its time expansion be constructed, then a reduced unfolding is obtained from \(Unf(N_X)\) in the following four steps:

1. Remove all the places \(p^c\) and the place \(p^t\) from the unfolding and all the incidental arcs.
2. Add the names \((t,b)\) and tic to \(T_{(t,b),M}\) and tic(M) respectively.
3. Delete all arcs \((p,tic(M))\) if \(C(p)\) is an untimed set.
(4) All consequent tic sequences \( \text{tic}(M) < \text{tic}(M) < \ldots < \text{tic}(M) \) should be changed by one \( \text{tic}_N \), where \( N \) is the length of the sequence. The reduced unfolding is denoted by \( R(\text{Unf}(N_X)) \).

From the definition it follows that the configuration \( C = (t_1, \ldots, t_n) \) of \( \text{Unf}(N_X) \) has a corresponding configuration \( C' = (t_1', \ldots, t_m') \) of \( R((\text{Unf}(N_X))) \), such that \( m \leq n \) and

- if \( t_i = T_{M,(t,b)} \), then \( \exists t_j' = (t,b) \) and \( j \leq i \);
- if \( t_i = \text{tic}(M) \), then \( \exists t_j' = \text{tic} \) and \( j \leq i \);

\[ \text{Mark}_{\text{Unf}(N_X)}(C) \cap P = \text{Mark}_{R(\text{Unf}(N_X))}(C') \].

Now, having a usual CPN as the time expansion of TCPN, we can construct \( R(\text{Unf}(X(TCPN))) \) as the unfolding of TCPN. Unfortunately, the obtained time expansion is not \( n \)-safe and we can’t guarantee finiteness of the generated unfolding. The problem is the same as for the occurrence-graph (O-graph) of TCPN. Like for the O-graph, we can propose two methods.

1. We can construct a partial unfolding by restricting the allowed time value. Formally this can be done by defining the colours \( \mathcal{A} = \text{int with } 0..N \) for timed \( C(p) \) and the place \( p^1 \). In this case we obtain S-finite CPN and accordingly to theorem 1 its unfolding is finite.

2. We can define some equivalence specification concerned with time values using the technique proposed in [10] in order to make the unfolding finite.

As in the previous part, the exact algorithm of \( R(\text{Unf}(X(TCPN))) \) generation is not given here. Instead we just give the idea of how to construct \( R(\text{Unf}(X(TCPN))) \).

The net example and its unfolding are given below.

![Fig. 5 The Dining Philosophers](image-url)
Now we show how to modify the unfolding algorithm for CPN into the algorithm of constructing the reduced unfolding for a given TCPN. First we should store the time values for each place (token element) in the generated part of unfolding. At each step of considering a co-set of places to be a preset for a given binding element \((t,b)\), we should also calculate the number and values for all tic-transitions such that \(\text{tic} < P'\), where \(P'\) are the places from the current co-set. This takes \(O(N_T)\) time. Let us notice that in the case of interval-timed CPN it takes \(O(N_T N_P)\). So we should multiply complexity of the unfolding algorithm by \(N_T\).

Finally we don’t need any special deadlock technique for TCPN, since the tic transition is not possible in the “colour deadlock”. We can apply a usual defini-
tion of a deadlock as a state \((M,r)\) for which there is no enabled tic- or occur-
transition. Accordingly to theorem 3, we have that \((M,r)\) is a deadlock in TCPN
\(\Leftrightarrow X(M,r)\) is a deadlock in its time expansion. Otherwise, applying the equiva-
lences from theorem 3 (1) and (2), we would come to contradiction with the
deadlock definition. While reduction of an unfolding doesn’t remove any transi-
tion, we have the correspondence between the deadlocks found in
Unf\((X(TCPN))\) and the deadlocks found in R(Unf\((X(TCPN))\)).

**CONCLUSION**

In paper [10] the unfolding technique proposed by McMillan in [12] and de-
veloped in later works has been applied to coloured Petri nets as they are de-
scribed in [8,9]. The technique has been formally transferred, two algorithms and
three finitization criteria have been considered.

The size of unfolding is often much smaller than the size of the reachability
graph of a PN. Using the EQ-cutoff criterion and symmetry or equivalence
specifications in unfolding generation [10], we can additionally reduced the size
of unfolding.

This paper transfer the unfolding technique from [1] to interval-timed CPN
and also considers unfolding generation for timed CPN as they are described in
[8,9]. Let us notice here that the notion of unfolding with equivalence given in
[10] is very useful when we want to obtain the complete unfolding of TCPN con-
sidered in [8,9].

In the future it is planned to make all the necessary experiments with unfold-
ings of coloured Petri nets.

**Acknowledgments.** I would like to thank Dr. Valery Nepomniaschy for
drawing my attention to this problem and Dr. Elena Bozhenkova for valuable
remarks.

**REFERENCES**

1. **Bieber B., Fleischhack H.** Model Checking of Time Petri Nets Based on Partial Order Semantics

2. **Cheng A., Christensen S., Mortensen K. H.** Model Checking Coloured Petri Nets Exploiting

3. **Couvreur J.-M., Grivet S., Poitrenaud D.** Designing an LTL Model-Checker Based on Unfold-
В.Е. Козюра

РАЗВЕРТКИ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ СО ВРЕМЕНЕМ

Препринт
82