

A New Graph of Classes for the Preservation of Quantitative Temporal Constraints

Xiaoyu Mao¹, Janette Cardoso², and Robert Valette³

¹ IRIT-UT1/LAAS, Toulouse, France,

² IRIT-UT1, 21, allées de Brienne, 31042 Toulouse, France

³ LAAS-CNRS, 31077 Toulouse, France

xmao@etud.insa-toulouse.fr, jcardoso@univ-tlse1.fr, robert@laas.fr

Abstract. The objective of this paper is to present a new abstract state space for t-time Petri nets which associates with each path in this space a sequence effectively firable in the net. This means that this state space has to exactly (in a quantitative way) define the set of constraints which have to be verified by the firings. After some definitions about the Simple Temporal Networks, the abstract states are defined as well the generation of the abstract space. It is shown that this space does not coincide with the two previously defined spaces (W and A) in TINA.

1 Introduction

For checking some properties of critical embedded systems such as the timeliness property for correct environment interaction, it is frequently necessary to consider specific scenarios of operations and to analyze the temporal constraints which have to be verified by the events composing them [Ri 05].

Other properties (related for example to the fact that a state is not reachable) imply the exhaustive search for all the states of a system. When temporal constraints exist, the states, in an infinite number, can be covered by a finite set of state classes for bounded Petri nets. In this case, a graph of state classes can be built in order to study the system, where nodes are state classes and the arc from a class \mathcal{C} to a class \mathcal{C}' is labeled by the transition t (leading from \mathcal{C} to \mathcal{C}'). Several kinds of classes have been proposed according to the kind of properties to be proven (properties expressed in LTL or in CTL for instance) [Me 85, Yo 98, Be 04, Ca 05]. Some approaches allow deriving the temporal constraints associated with a given scenario, directly from the Petri net [PR 99, Ri 01, Ri 05]. However, they cannot be used efficiently for t-time Petri nets with strong semantics and interleaving. In order to correctly delimit the domains of the variables attached to the firing dates in a transition firing sequence, it is necessary, in the case of a t-time Petri net with strong semantics, to know the transition enabling dates. This implies that, for each transition, the date of the firing which has produced the last token is known. In consequence, it is necessary to proceed in the context of interleaving semantics and therefore to explicitly consider states and firing sequences (in contrast with [PR 99, Ri 01] where the approach is based on scenarios i.e. partial orders).

It is clear that it is always possible, given a firing sequence possibly derived from a graph of classes, to obtain a set of constraints delimiting the firing dates by considering both the Petri net and the graph of classes [Sc 04]. In this paper, the proposed approach is to construct a graph of classes with sets of constraints attached to its arcs, such that the constraints which have to be verified by the firing dates for any sequence in the net, are directly derived by concatenating the constraints attached to the arcs covered by the corresponding sequence in the graph of classes.

2 Basic Notions

2.1 Simple Temporal Network

Definition 1 (Simple temporal network). A simple temporal network N is composed of a finite set V of variables v_i and a finite set C of **binary** constraints $C_{ij}(v_i, v_j)$ defined as convex intervals $[c_{mij}, c_{Mij}]$ delimiting the possible distance between two variables v_i and v_j of V . Each C_{ij} is therefore equivalent to: $c_{mij} \leq v_j - v_i \leq c_{Mij} \quad v_i, v_j \in V$.

Definition 2 (Complete network). A simple temporal network N is complete iff a constraint C_{ij} is associated with each pair of variables.

Definition 3 (Minimal network). A complete simple temporal network $N = (V, C)$ is minimal iff $\forall v_i, v_j \in V$ and $\forall c \in C_{ij}$ ($c_{mij} \leq c \leq c_{Mij}$), there is an assignment of values to all the variables of V which verifies all the constraints and such that $v_j - v_i = c$.

The Floyd-Warshall algorithm, derives a complete and minimal simple temporal network from any consistent (having at least one solution) simple temporal network [De 91, Gh 04]. After applying this algorithm, the new resulting constraint $C_{ij} = [d_{mij}, d_{Mij}]$ is such that d_{mij} is the best possible lower bound, d_{Mij} is the best possible upper bound and these bounds are actually reached for at least one set of assignments of all the variables of V verifying all the constraints of C .

Definition 4 (Intersection). The intersection of two simple temporal networks $N = (V, C)$ and $N' = (V', C')$ is the simple temporal network $N'' = N \cap N' = (V'', C'')$ such that: *i*) $V'' = V \cap V'$, *ii*) $\forall v_i, v_j \in V''$, $C''_{ij} = C_{ij} \cap C'_{ij}$.

The constraints are given by the intersection of the intervals. If one of these intersections is empty, the simple temporal network N'' is inconsistent in an obvious way.

Definition 5 (Union). Let us consider two temporal networks $N = (V, C)$ and $N' = (V', C')$ such that: $\forall v_i, v_j \in V \cap V'$, $C_{ij} = C'_{ij}$. The union $N'' = N \cup N'$ is a simple temporal network (V'', C'') such that: *i*) $V'' = V \cup V'$, *ii*) $C'' = C \cup C'$.

It has to be pointed out that as for any pair of variables v_i and v_j belonging both to N and N' , the constraints C_{ij} and C'_{ij} are the same in N and N' , and the union of the sets of constraints C and C' is always consistent.

2.2 t-Time Petri Nets

Definition 6. A *t-time Petri Net* [Be 91, Be 04] is a 3-tuple $\langle \mathcal{N}, M_0, I \rangle$ where:

- $\mathcal{N} = \langle P, T, Pre, Post \rangle$ is a Petri net,
- M_0 : is the initial marking,
- $I : T \rightarrow (Q^+ \cup 0) * (Q^+ \cup \infty)$ is the static interval function.

The static interval function I associates with each transition t_i a temporal interval $[a_i, b_i]$ (see figure 1) that represents the set of its possible firing dates counting from its enabling date.

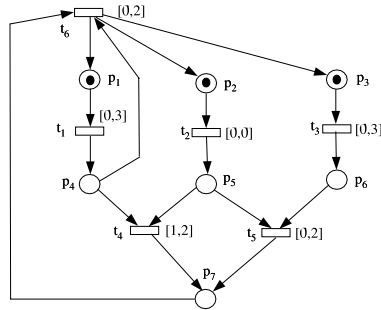


Fig. 1. Example of a t-time Petri net

Typically, for t-time Petri nets, the operational semantics includes the so-called *strong semantics* which enforces the firing of one of the enabled transitions before the earliest of all the latest firing dates for the enabled transitions. This means that a transition cannot remain enabled without being fired after the end of its firing interval. In this paper, it is assumed that there is no memory of the enabling time of a transition in the past and transitions may be enabled concurrently.

In a t-time Petri Net, the following events associated with a transition (and the corresponding temporal variables) must be taken into account: the *enabling date*, *begin/end of the firing interval* and *firing date*. The following constraints must be verified between the variables corresponding to these events:

- the enabling date of a transition is equal (not greater) to the firing date of the last transition which has contributed to its enabling,
- the transition firing date should be included in its firing interval I .

These relations can be expressed by simple binary constraints when the operational semantics is such that only firing sequences (totally ordered) are considered (interleaving semantics). In consequence, simple temporal networks are an adequate framework to analyze the temporal constraints generated by t-time

Petri nets. In the following, only two types of variables are considered: x_i^k which is the variable denoting the date of the k^{th} firing of transition t_i and y_i which denotes the upper bound of the firing interval of enabled transition t_i .

3 Definition of States and State Classes

3.1 State of a t-Time Petri Net

Let us consider the execution of a firing sequence $\sigma = t_1; \dots; t_i; t_j; \dots; t_n$ in a t-time Petri net. A transition can be fired several times in a sequence. We consider a firing of transition t_i which is the oi^{th} firing of this transition and the next firing in σ is the oj^{th} firing of t_j . The corresponding variables are x_i^{oi} for t_i and x_j^{oj} for t_j .

Given a specific execution of σ , the state after the firing of t_i is the obtained marking associated with the current clock value and the firing dates of all the transitions preceding t_i in σ in order to compute the remaining firing intervals for each enabled transition.

3.2 State Classes

A class is composed of all the states which are reachable by an execution of σ after the firing of t_i and before that of t_j . The class has to allow the accurate definition of all the constraints which have to be verified by the firing date of transitions t_j and by the following ones in σ . This means that it is necessary to be able to derive not only the distance of x_j^{oj} and x_i^{oi} , but also the distance of x_j^{oj} with all the preceding firing dates in σ .

In order to have a finite number of classes, it is necessary to *forget* a part of the past. Instead of keeping all the variables corresponding to the past transition firings and the corresponding simple temporal network, it is possible to only keep a fragment of it.

After having defined this fragment, the paper gives the procedure of construction of the temporal constraints which variable x_j^{oj} must verify. These constraints are attached to the arcs of the graph of classes under the form of a simple temporal network. Then it is proven that the fragment is sufficient, *i.e.* that considering more variables and more constraints about the past events would not modify the simple temporal networks attached to the arcs.

Definition 7. *The initial state class \mathcal{C}_0 is defined by the tuple (M_0, N_{c_0}) where:*

- M_0 is the initial marking of the net; it is assumed that n_0 transitions are enabled by M_0 ,
- N_{c_0} is the temporal network composed only of the variable x_0 representing the time origin.

The time origin is the event that has enabled all n_0 enabled transition at class \mathcal{C}_0 and in a certain way, it is the *beginning of the world*.

Let σ be a firing sequence $t_1; \dots; t_i$ of a t-time Petri net, t_i the last fired transition in σ and $t_{s(k)}$ the transition that has enabled a transition t_k .

Definition 8. *The state class \mathcal{C} , obtained after the firing of transition t_i , is defined by the pair $\{M, Nc\}$ where:*

- M is the current marking of the net; it is assumed that n transitions are enabled by M ,
- Nc is the minimal and complete simple temporal network composed of the following variables and constraints :
 1. the variable x_i^{oi} associated with the last transition firing (t_i firing),
 2. for each enabled transition t_k from M , the variable associated with the firing of transition $t_{s(k)}$ which has enabled t_k , $x_{s(k)}^{osk}$ ($k = 1, \dots, n$),
 3. the temporal constraints between these variables (minimal and complete network).

If \mathcal{C}_p is the class from which transition t_i has been fired, \mathcal{C} is the class obtained by the firing of t_i at date x_i^{oi} and Nt_{i,c_p} the simple temporal network delimiting t_i firing, figure 2 describes the relationships between classes and simple temporal networks.

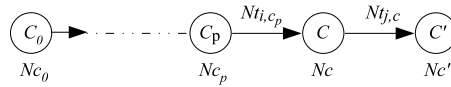


Fig. 2. A piece of a graph of classes

For all classes $\mathcal{C} \neq \mathcal{C}_0$, the representation of the past includes the last transition firing and the firing which has enabled each one of the n enabled transitions at \mathcal{C} . There are two cases:

- two transitions t_1 and t_2 have been enabled by the same transition t_a , the corresponding variables $x_{s(1)}^{os1}$ and $x_{s(2)}^{os2}$ are the same, $x_{s(1)}^{os1} = x_{s(2)}^{os2} = x_a^{oa}$;
- the last transition firing (represented by variable x_i^{oi}) is also the event that enables a transition t_j in this class (represented by variable $x_{s(k)}^{osk}$), these variables are the same, $x_i^{oi} = x_{s(k)}^{osk}$.

In the sequel if a transition t_i appears only once in a sequence σ ($o_i=1$), the corresponding variable is noted x_i instead of x_i^1 .

For example, let us consider the t-time Petri net in figure 1, with the firing of sequence $t_2; t_3; t_1$ from initial class \mathcal{C}_0 with initial marking $p_1p_2p_3$ and the temporal network Nc_0 given by x_0 . The class \mathcal{C} reached by the firing of this sequence has the marking $p_4p_5p_6$. The transitions enabled by this marking are t_4 and t_5 . The following events must be considered in order to construct Nc :

- the last transition fired in the sequence is $t_i = t_1$, and the corresponding variable is x_1 ;
- t_4 has been enabled by t_1 firing, so $s(4)=1$, and $x_{s(4)}^{os4} = x_1$;
- t_5 has been enabled by t_3 firing, so $s(5)=3$, and $x_{s(5)}^{os5} = x_3$.

The complete definition of the temporal network Nc requires the constraint values associated with variables x_1 and x_3 . Nc is a fragment of the temporal network delimiting the last transition firing (t_1 in this example). The definition of the temporal network Nt delimiting the firing of a transition t is defined in the sequel.

3.3 The Temporal Network Delimiting the Firing of t_j

In a reachable marking graph obtained from a classical Petri net (without temporal information), a node corresponds to a marking and an arc between two nodes (n_1, n_2) is labeled by the transition whose firing leads from n_1 to n_2 . In the graph of classes obtained from a t-time Petri net, an arc between two classes must be labeled, besides the transition, by temporal information delimiting the firing date of this transition. According to the definition of class and the corresponding definition of temporal information attached to the arc, several graphs of classes have been proposed allowing to prove different properties of a t-time Petri net ([Be 91, Yo 98], etc). In our approach, an arc, labeled by a transition t_j , is also associated with a temporal network $Nt_{j,i}$ delimiting the firing date of t_j from \mathcal{C}_i . These constraints reflect the memory of the past necessary to characterize the future events.

Let t_j be a transition among the n enabled transitions at class $\mathcal{C} = (M, Nc)$ (def. 8), with $Nc = (Vc, Cc)$, and let $t_l, l \neq j$ be the other $n - 1$ enabled transitions at \mathcal{C} .

Definition 9. *The **simple temporal network** $Nt_{j,c} = (Vt, Ct)$ delimiting the firing of t_j from class \mathcal{C} is composed of the following variables and constraints:*

1. all variables and constraints from $Nc, Vt = Vc, Ct = Cc$,
2. the variable x_j^{oj} (firing date of t_j) and the static interval $I(t_j)$ as a constraint between x_j^{oj} and $x_{s(j)}^{osj}$,
3. the variable y_l^{ol} corresponding to each enabled transition t_l and the singleton $[d_{Ml}, d_{Ml}]$ (the upper bound of static interval $I(t_l)$) as a constraint between $(x_{s(l)}^{osl}, y_l^{ol})$,
4. the constraint $[0, \infty[$ between x_i^{oi} and x_j^{oj} to express the fact that t_j must be fired after t_i ,
5. the constraint $[0, \infty[$ between the pairs $(x_j^{oj}, y_l^{ol}), l \neq j$, to express the fact that t_j must be fired before the upper bound of the firing interval of transitions t_l .

The minimal and complete $Nt_{j,c}$ is obtained after applying Floyd-Warshall algorithm. All variables y_l^{ol} and the constraints to which they are directly connected can be deleted (proved in section 3.4).

Step 1 indicates that Nc is a fragment of $Nt_{j,c}$, $Nc \subseteq Nt_{j,c}$. By the way, variables $x_{s(j)}^{osj}$ and $x_{s(l)}^{osl}$ in Steps 2 and 3 respectively belong to Nc (def. 8), since they correspond to transitions which have enabled one of n enabled transitions in \mathcal{C} . Step 4 is imposed by the interleaving semantics and step 5 is imposed by the strong semantics. If the network is not consistent, it means that t_j cannot be fired before the other $n - 1$ transitions t_l .

Let us consider the initial class \mathcal{C}_0 of the net in fig. 1, with $M_0 = p_1p_2p_3$ and $Nc_0 : x_0$. Let us consider the firing of t_2 (at date x_2), delimited by $Nt_{2,0}$. At the beginning, $Nt_{2,0} = Nc_0 : x_0$ (step 1); node x_2 and arc $(x_0, x_2)=[0\ 0]$ are added ($s(2) = 0$, step 2). As t_1 and t_3 are also enabled at \mathcal{C}_0 , nodes y_1 and y_3 as well arcs $(x_0, y_1)=(x_0, y_3)=[3\ 3]$ are added ($s(1) = s(3) = 0$, step 3). Arc $(x_0, x_2)=[0\ \infty[$ ($i = 0$, step 4) is also added, as well $(x_2, y_1)=(x_2, y_3)=[0\ \infty[$ (step 5), leading to $Nt_{2,0}$ of fig. 3.a. After Floyd-Warshall algorithm, the obtained $Nt_{2,0}$ is the one of fig. 3.b. The final $Nt_{2,0}$ is represented by the dotted arc of this figure.

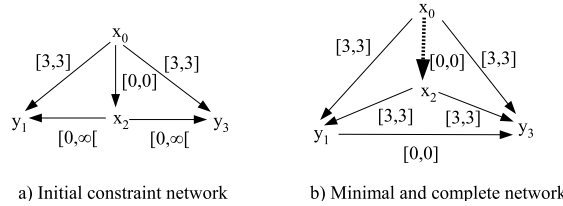


Fig. 3. Temporal network $Nt_{2,0}$ delimiting t_2 firing from \mathcal{C}_0

It is important to remark that the simple temporal network Nc' of the class \mathcal{C}' reached after the firing of t_j is also included in $Nt_{j,c}$. In fact, all n' enabled transitions at \mathcal{C}' have been enabled by the firing of t_j ($x_j \in Nt_{j,c}$) or the firing of a previous transition t in the sequence leading to \mathcal{C} ($x \in Nc \subseteq Nt_{j,c}$).

3.4 Proofs

Proving that y_l^{ol} can be Deleted. After Floyd-Warshall execution, variables $x_{s(l)}^{osl}$ and y_l^{ol} are redundant because the constraints connecting them are singletons. Indeed knowing the constraint connecting $x_{s(l)}$ (event that has enabled transition t_l) to x_k is sufficient to derive the triangle in figure 4.a, because:

$$d_{mkl} = d_M - d_{msk} \quad \text{and} \quad d_{Mkl} = d_M - d_{Msk} \tag{1}$$

Proving that the Past can be Forgotten. Let us consider the class \mathcal{C}' obtained from the firing of t_j , characterized by the temporal network Nc' in fig. 4.b, with a variable x_k belonging to the forgotten past. The proof that a

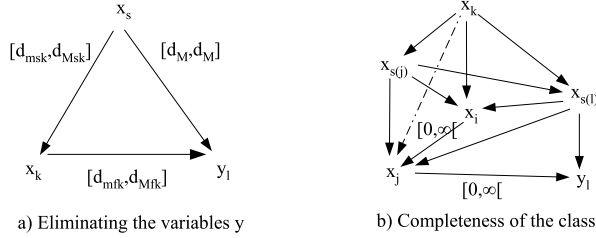


Fig. 4. Temporal network Nt illustrating the proof

part of the past can be forgotten, upon which the definition of class is based, relies on the assumption that x_k cannot restrict the constraints between x_j and the other nodes.

As underlined in section 3.3, the temporal network $Nt_{j,c}$ (delimiting the firing of a transition t_j from a class \mathcal{C}) is constructed from the temporal network Nc of the class \mathcal{C} (obtained from the firing of a precedent transition t_i in the sequence).

So, if the variables y_l are not deleted in step 7 of the procedure given in section 3.3, the obtained class \mathcal{C}' is composed by 3 kind of constraints: 1) between x_j and $x_{s(j)}$ (event that has enabled transition t_j), 2) between x_j and x_i (the last transition fired before t_j , interleaving semantics) and 3) between x_j and y_l , $l \neq j$ (strong semantics). Only y_1 is represented in the figure, the other nodes y_l are identical. The constraints C_{kj} is initially equal to $[0, \infty[$.

The longest path of x_k towards x_j thus passes necessarily by $x_{s(j)}$, x_i or one of nodes $x_{s(l)}$. Let us suppose that it is going through $x_{s(j)}$, so $d_{mkj} = d_{mks(j)} + d_{ms(j)j}$ and $d_{Mkj} = d_{Mks(j)} + d_{Ms(j)j}$. Can this constraint restricts $[d_{mij}, d_{Mij}]$, for example? (An arc $(x_i, x_j) = [d_{mij}, d_{Mij}]$ corresponds to two arcs $(x_i, x_j) = d_{Mij}$ and $(x_j, x_i) = -d_{mij} = d_{mji}$.) In this case:

$$d_{mij} = d_{mik} + d_{mkj} = d_{mik} + d_{mks(j)} + d_{ms(j)j} \tag{2}$$

$$d_{Mij} = d_{Mik} + d_{Mkj} = d_{Mik} + d_{Mks(j)} + d_{Ms(j)j} \tag{3}$$

As the temporal network Nc characterizing the class \mathcal{C} is complete and minimal, $d_{mis(j)} \geq d_{mik} + d_{mks(j)}$ and $d_{Mis(j)} \leq d_{Mik} + d_{Mks(j)}$, so the lower bound of the path going directly through $x_{s(j)}$ is equal or *bigger* than the one going through x_k (eq. 2), and the upper bound is equal or *smaller* than it (eq. 3).

The other cases are analogous. Let us take into account the case where the longest path between x_j and x_k goes through $x_{s(j)}$ and the one between x_k and y_l goes through $x_{s(l)}$. If x_k can reinforce the constraint between x_j and y_l :

$$d_{mjl} = d_{mjk} + d_{mkl} = d_{mjs(j)} + d_{ms(j)k} + d_{mks(l)} + d_{ms(l)l} \tag{4}$$

$$d_{Mjl} = d_{Mjk} + d_{Mkl} = d_{Mjs(j)} + d_{Ms(j)k} + d_{Mks(l)} + d_{Ms(l)l} \tag{5}$$

But $d_{ms(j)s(l)} \geq d_{ms(j)k} + d_{mks(l)}$ and $d_{Ms(j)s(l)} \leq d_{Ms(j)k} + d_{Mks(l)}$, so the lower bound of the path $(x_j, x_{s(j)}, x_{s(l)}, y_l)$ is equal or *bigger* than the one going through x_k , and the upper bound is equal or *smaller* than it.

It is proved that the firing date x_k do not constraint the distance between other nodes, so it can be forgotten in the network construction.

3.5 Restricted Class

The constraints between the events of a class \mathcal{C} are obtained from the set of constraints having to be checked by the transition firing leading to \mathcal{C} . Indeed, the temporal network Nc of class \mathcal{C} , reached from t_i firing, is a fragment of the network Nt_{i,c_p} characterizing the firing date of t_i (from a previous class \mathcal{C}_p).

During the construction of a temporal network $Nt_{j,c}$ from the class \mathcal{C} , some constraint $C_{k,l}$ between two nodes x_k and x_l can become more restricted than its initial value in the network Nc of \mathcal{C} . This means that transition t_j can only be fired from the states of \mathcal{C} for which variables x_k and x_l verify this new, more restricted constraint $C_{k,l}$. This defines a sub-class $\mathcal{C}r^j$ restricted in order to permit the firing of t_j .

Let $\mathcal{C} = (M, Nc)$ be a class, let t_j be a transition which can be fired from \mathcal{C} and whose firing date is delimited by the $Nt_{j,c}$. If $Nt_{j,c} \cap Nc \neq Nc$ (see definition 4) then it is necessary to define a restricted class.

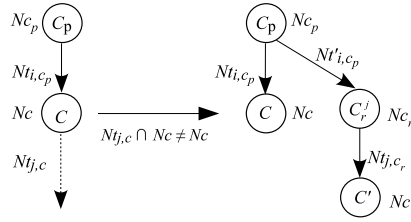


Fig. 5. Restricted class $\mathcal{C}r$ of class \mathcal{C}

Definition 10. The restricted class $\mathcal{C}r^j = (M_r, Nc_r)$ of class \mathcal{C} is defined by

- $M_r = M$,
- Nc_r is the network $Nt_{j,c} \cap Nc$ after application of Floyd-Warshall

The class $\mathcal{C}r^j$ characterizes the states from where the transition t_j is fireable; these states have been reached by the firing of t_i (preceding t_j in the sequence $\tau = t_1; \dots; t_i; t_j$) whose firing date has to be consistent with $\mathcal{C}r^j$ (fig. 5). This means that this date is delimited by the network $Nt'_{i,c_p} = Nt_{i,c_p} \cap Nc_r$, which, after a new application of Floyd-Warshall may be such that $Nt'_{i,c_p} \cap Nc_p \neq Nc_p$ and require so a restricted class \mathcal{C}^i_{pr} for \mathcal{C}_p and so on.

3.6 Equivalent Classes

Definition 11. Two classes $\mathcal{C} = (M, Nc)$ and $\mathcal{C}' = (M', Nc')$, differing from the initial class \mathcal{C}_0 , with $Nc = (X, C)$ and $Nc' = (X', C')$ are equivalent if:

1. they have the same marking, $M = M'$,
2. there exists a bijection τ between the elements of X and X' such that
 - $x'_k = \tau(x_i)$ implies $k = i$ (the variables are firing dates of the same transition)
 - if $x'_i = \tau(x_i)$ and $x'_j = \tau(x_j)$ then C'_{ij} (constraint between x'_i and x'_j) is equal to C_{ij} (constraint between x_i and x_j)

Definition 12. A class $\mathcal{C} = (M, Nc)$ is equivalent to the initial class $\mathcal{C}_0 = (M_0, x_0)$ if:

1. they have the same marking, $M = M_0$,
2. the set of variables X of Nc is a singleton, $X = \{x_k\}$ (no past memory).

In the above definition, the firing of transition t_k leads the system back to the initial marking M_0 and enables all transitions at this marking. That is why it is equivalent to the *beginning of the world*.

3.7 Graph of Classes

The graph of classes is composed by classes $\mathcal{C} = (M, Nc)$ (def. 7 and 8) and the arcs connecting them. An arc $(\mathcal{C}, \mathcal{C}')$ is labeled by the transition t (leading from \mathcal{C} to \mathcal{C}') and $Nt_{i,c}$ delimiting this firing (section 3.3). The graph generator (in Java) can be downloaded at <http://www.irit.fr/~Janette.Cardoso/feria> as well the algorithm description ([AlgoGraphC.pdf](#)) and the manual ([readme](#)).

3.8 Sequence Characterization

The temporal network of a sequence $\sigma = t_1; \dots; t_i; t_j; \dots; t_n$ from a class \mathcal{C} is given by the union (see definition 5) of the temporal networks delimiting each transition firing in σ : $Nt_\sigma = Nt_{1,c} \cup \dots \cup Nt_{i,c_i} \cup Nt_{j,c_j} \cup \dots \cup Nt_{n,c_n}$.

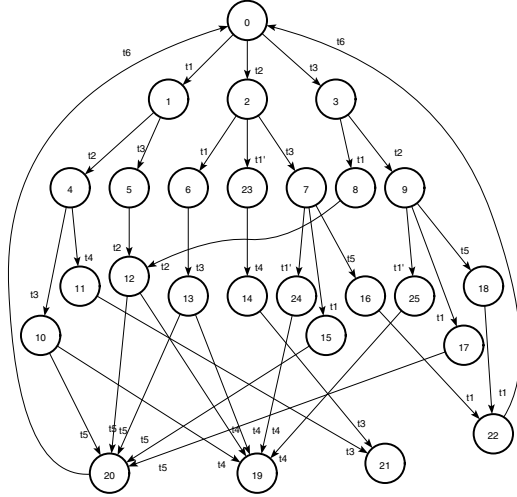
It must be pointed out that this expression is consistent with the union of simple temporal networks because by construction $Nt_{i,c_i} \cap \mathcal{C}_j = \mathcal{C}_j$, $Nt_{j,c_j} \cap \mathcal{C}_j = \mathcal{C}_j$ and $Nt_{i,c_i} \cap Nt_{j,c_j} = \mathcal{C}_j$. This means that the constraints between variables belonging both to Nt_{i,c_i} and Nt_{j,c_j} are equal, they are those of \mathcal{C}_j .

A same sequence σ can be associated with more than one path in the graph of classes. A particular case appear when $\sigma = t_1; \dots; t_i$ and the firing of last transition t_i leads to a class \mathcal{C} and also to its restricted class \mathcal{C}_r . It means that \mathcal{C}_r was created in such a way that another transition t_j can be fired after t_i . Two network are obtained, N_σ^1 leading to \mathcal{C} and N_σ^2 leading to \mathcal{C}_r , but $N_\sigma^1 \supseteq N_\sigma^2$ and so the network characterizing σ is N_σ^1 (see the example in section 4.2).

4 Example

4.1 Construction of the Graph of Classes

Let us consider the example of figure 1, presenting a deadlock and also infinite sequences. Table 1 shows the classes (marking and temporal network Nc) obtained with the proposed approach and figure 6 represents the graph of classes.



Temporal networks $Nt_{i,k}$ of t_i from C_k	
$Nt_{1,0}(x_0, x_1)=[0\ 0]$	$Nt_{2,0}(x_0, x_2)=[0\ 0]$
$Nt_{3,0}(x_0, x_3)=[0\ 0]$	
$Nt_{2,1}(x_0, x_1)=(x_0, x_2)=(x_1, x_2)=[0\ 0]$	
$Nt_{3,1}(x_0, x_1)=(x_0, x_3)=(x_1, x_3)=[0\ 0]$	
$Nt_{1,2}(x_0, x_2)=[0\ 0], (x_0, x_1)=(x_2, x_1)=[0\ 3]$	
$Nt_{3,2}(x_0, x_2)=[0\ 0], (x_0, x_3)=(x_2, x_3)=[0\ 3]$	
$Nt_{1,3}(x_0, x_3)=(x_0, x_1)=(x_3, x_1)=[0\ 0]$	
$Nt_{2,3}(x_0, x_3)=(x_0, x_2)=(x_3, x_2)=[0\ 0]$	
$Nt_{3,4}(x_0, x_2)=[0\ 0], (x_0, x_3)=(x_2, x_3)=[0\ 2]$	
$Nt_{4,4}(x_0, x_2)=[0\ 0], (x_0, x_4)=(x_2, x_4)=[1\ 2]$	
$Nt_{2,5}(x_0, x_3)=(x_0, x_2)=(x_3, x_2)=[0\ 0]$	
$Nt_{3,6}(x_0, x_1)=(x_0, x_3)=[0\ 3], (x_1, x_3)=[0\ 2]$	
$Nt_{1,7}(x_0, x_3)=(x_0, x_1)=[0\ 3], (x_3, x_1)=[0\ 2]$	
$Nt_{5,7}(x_0, x_3)=(x_0, x_5)=[0\ 3], (x_3, x_5)=[0\ 2]$	
$Nt_{2,8}(x_0, x_1)=(x_0, x_2)=(x_1, x_2)=[0\ 0]$	
$Nt_{1,9}(x_0, x_2)=[0\ 0], (x_0, x_1)=(x_2, x_1)=[0\ 2]$	
$Nt_{5,9}(x_0, x_2)=[0\ 0], (x_0, x_5)=(x_2, x_5)=[0\ 2]$	
$Nt_{4,10}(x_2, x_3)=(x_3, x_4)=[0\ 2], (x_2, x_4)=[1\ 2]$	
$Nt_{5,10}(x_2, x_3)=(x_0, x_1)=(x_2, x_1)=[0\ 2]$	
$Nt_{4,12}(x_2, x_4)=[1\ 2], Nt_{6,12}(x_2, x_5)=[0\ 2]$	
$Nt_{4,13}(x_1, x_3)=(x_3, x_4)=[0\ 2], (x_1, x_4)=[1\ 2]$	
$Nt_{5,13}(x_1, x_3)=(x_1, x_5)=(x_3, x_5)=[0\ 2]$	
$Nt_{3,14}(x_0, x_4)=(x_0, x_3)=[1\ 3], (x_4, x_3)=[0\ 2]$	
$Nt_{5,15}(x_0, x_1)=(x_0, x_5)=(x_1, x_5)=[0\ 2]$	
$Nt_{1,16}(x_0, x_5)=(x_0, x_1)=(x_5, x_1)=[0\ 3]$	
$Nt_{5,17}(x_2, x_1)=(x_2, x_5)=(x_1, x_5)=[0\ 2]$	
$Nt_{1,18}(x_0, x_5)=[0\ 2], (x_0, x_1)=(x_5, x_1)=[0\ 3]$	
$Nt_{6,20}(x_5, x_6)=[0\ 2] Nt_{6,22}(x_1, x_6)=[0\ 2]$	
$Nt'_{1,2}(x_0, x_2)=[0\ 0], (x_0, x_1)=(x_2, x_1)=[0\ 2]$	
$Nt_{4,24}(x_3, x_1)=[0\ 1], (x_3, x_4)=(x_1, x_4)=[1\ 2]$	
$Nt'_{1,7}(x_0, x_3)=(x_0, x_1)=[0\ 3], (x_3, x_1)=[0\ 1]$	
$Nt_{4,25}(x_2, x_1)=[0\ 1], (x_2, x_4)=(x_1, x_4)=[1\ 2]$	
$Nt'_{1,9}(x_0, x_2)=[0\ 0], (x_0, x_1)=(x_2, x_1)=[0\ 1]$	

Fig. 6. a) Graph of classes preserving firing constraints, b) Temporal networks on arcs

Let us consider the sequence $\sigma_1 = t_2; t_3; t_1; t_4$ in the graph of figure 6, leading to a deadlock $M_{19} = p_6p_7$. The initial class C_0 (see table 1) is defined by the initial marking $M_0 = p_1p_2p_3$ and the network Nc_0 composed by a unique node x_0 corresponding to an initial event creating the tokens of the initial marking. It represents the time origin.

Considering the firing of t_2 (at the date x_2) from C_0 , the final temporal network $Nt_{2,0}$ delimiting this firing is represented by the dotted arc in fig. 3.b (as explained in section 3.3. The network Nc_2 of the new class C_2 , reached with t_2 firing (obtained from $Nt_{2,0}$) is also represented by the dotted arc in fig. 3.b.

Transitions t_1 and t_3 are enabled at C_2 . Let us consider the firing of t_3 . The network $Nt_{3,2}$ delimiting t_3 firing (fig. 7.a) brings the system from the class C_2 to class C_7 . The networks Nc_2 and Nc_7 are represented in figure 7.a respectively by C_{02} (the bold arc) and C_{03} (the dotted arc).

Let us consider now the firing of t_1 from C_7 . The minimal and complete network $Nt_{1,7}$ delimiting the firing date x_1 is given by fig. 7.b. The reached class is C_{15} (bold dotted arc in fig. 7.b).

The firing of t_4 from C_{15} is defined by $Nt_{4,15}$ delimiting the firing date x_4 (fig. 7.c). The minimal network $Nt_{4,15}$ has the constraint $C'_{31} = [0\ 1]$ (on arc (x_3, x_1)), that is a reduced value in relation to that defined by Nc_{15} ($C_{31} = [0\ 2]$ or $0 \leq x_1 - x_3 \leq 2$) of class C_{15} . So, t_4 can be fired after t_1 only if t_1 is fired no more than $[0\ 1]$ unities of time after t_3 . The arc (x_3, x_1) defines a *restricted class* C_{24} , with $M_{24} = M_{15}$, and Nc_{24} given by $C'_{31} = [0\ 1]$. This class gathers all the states reached from the initial state by the firing of the sequence $t_2; t_3; t_1$ knowing that transition t_4 can be fired.

Table 1. Classes preserving the constraints

Class	Marking	Constraints Nc	Class	Marking	Constraints Nc
C_0	$p_1p_2p_3$	x_0	C_{13}	$p_4p_5p_6$	$0 \leq x_3 - x_1 \leq 2$
C_1	$p_2p_3p_4$	$0 \leq x_1 - x_0 \leq 0$	C_{14}	p_3p_7	$1 \leq x_4 - x_0 \leq 3$
C_2	$p_1p_3p_5$	$0 \leq x_2 - x_0 \leq 0$	C_{15}	$p_4p_5p_6$	$0 \leq x_1 - x_3 \leq 2$
C_3	$p_1p_2p_6$	$0 \leq x_3 - x_0 \leq 0$	C_{16}	p_1p_7	$0 \leq x_5 - x_0 \leq 3$
C_4	$p_3p_4p_5$	$0 \leq x_2 - x_0 \leq 0$	C_{17}	$p_4p_5p_6$	$0 \leq x_1 - x_2 \leq 2$
C_5	$p_2p_4p_6$	$0 \leq x_3 - x_0 \leq 0$	C_{18}	p_1p_7	$0 \leq x_5 - x_0 \leq 2$
C_6	$p_3p_4p_5$	$0 \leq x_1 - x_0 \leq 3$	C_{19}	p_6p_7	x_4
C_7	$p_1p_5p_6$	$0 \leq x_3 - x_0 \leq 3$	C_{20}	p_4p_7	x_5
C_8	$p_2p_4p_6$	$0 \leq x_1 - x_0 \leq 0$	C_{21}	p_6p_7	x_3
C_9	$p_1p_5p_6$	$0 \leq x_2 - x_0 \leq 0$	C_{22}	p_4p_7	x_1
C_{10}	$p_4p_5p_6$	$0 \leq x_3 - x_2 \leq 2$	C_{23}	$p_3p_4p_5$	$0 \leq x_1 - x_0 \leq 2$
C_{11}	p_3p_7	$1 \leq x_4 - x_0 \leq 2$	C_{24}	$p_4p_5p_6$	$0 \leq x_1 - x_3 \leq 1$
C_{12}	$p_4p_5p_6$	x_2	C_{25}	$p_4p_5p_6$	$0 \leq x_1 - x_2 \leq 1$

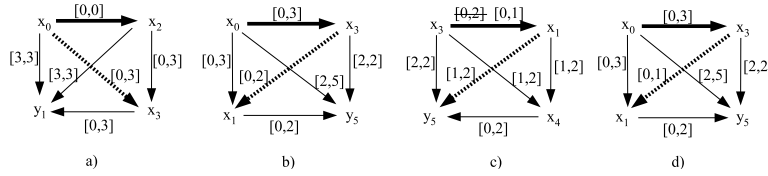


Fig. 7. Temporal networks a) $Nt_{3,2}$, b) $Nt_{1,7}$, c) $Nt_{4,15}$ and d) $Nt'_{1,7}$

Once the restricted class C_{24} was created, a new arc coming from the previous class C_7 to C_{24} must also be created, labeled by the temporal network delimiting the firing of t_1 (considering that transition t_4 can be effectively fired). This network is given by $Nt'_{1,7} = Nt_{1,7} \cap Nc_{24}$ (fig. 7.d). As $Nt_{3,2} \cap Nc_7 = Nc_{24}$, the backward propagation stops.

The class C_{15} is kept in the graph as well as the network $Nt_{1,7}$ labeling the arc (C_7, C_{15}) , but there is no output arc from C_{15} labeled by t_4 . Instead only the pair (C_{24}, t_4) is appended to the list of hanging nodes.

The firing of t_6 from classes C_{20} and C_{22} leads the system to a class C with $M = M_0$ and Nc given by x_6 (no past memory). Using definition 12, this class is equivalent to the initial class C_0 .

4.2 Temporal Network of a Sequence

Let us consider the sequence $\sigma_1 = t_2; t_3; t_1; t_4$ whose temporal network (fig. 8) is obtained by the union of $Nt_{2,0}$, $Nt_{3,2}$, $Nt'_{1,7}$ et $Nt_{4,24}$. Some arcs are represented twice in order to point out that they belong to two networks delimiting firing dates. There is indeed only one value and one constraint. For example, as $Nt'_{1,7} \cap Nt_{4,24} = Nc_{24}$, the arc (x_3, x_1) belongs to two networks and is represented by two dotted lines.

Let us now consider the sequence $\sigma_2 = t_2; t_3; t_1$. Two paths in the class graph and in consequence two networks are obtained: $N_{\sigma_2}^1 = Nt_{2,0} \cup Nt_{3,2} \cup Nt_{1,7}$

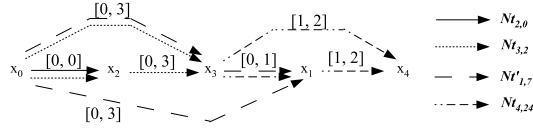


Fig. 8. Temporal network of the sequence σ_1

(leading to \mathcal{C}_{15}) and $N_{\sigma_2}^2 = N_{t_{2,0}} \cup N_{t_{3,2}} \cup N_{t'_{1,7}}$ (leading to \mathcal{C}_{24}). But $N_{\sigma_2}^1 \supseteq N_{\sigma_2}^2$, so the network characterizing σ_2 is given by $N_{\sigma_2}^1$.

The figure 9 shows the temporal network N_{σ_3} for the sequence $\sigma_3 = t_2; t_3; t_5; t_1; t_6; t_2; t_3$ where t_2 and t_3 fire twice in the sequence, and the second firings of t_2 and t_3 are represented by x_2^2 and x_3^2 respectively in the temporal network.

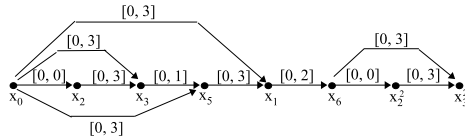


Fig. 9. Temporal network of the sequence σ_3

5 Related Work

Several approaches have been proposed to reduce the potentially infinite state spaces of real time systems to finite states spaces in order to analyze such systems [Me 85], [Be 91], [Yo 98], [Be 04]. All approaches are based on the equivalence of state classes. A first difference with these approaches is that, in the approach presented here, simple temporal network are used instead of geometrical region to deal with temporal information.

[Be 91] and [Be 04] have proposed the tool Tina, with two types of graph of classes, one is called Linear mode (W) and the other Atomic mode (A). They allow LTL and CTL property model checking, respectively. In the W mode, a class is given by its marking, the temporal domain of enabled transitions and the (non redundant) constraints existing between these transitions in the past. In the A mode, a class is given by its marking and a clock for each enabled transition (clock(t)=0 if t is newly enabled, otherwise it takes the previous value). Some classes in this mode correspond to a partition of the ones in W mode. Figure 10 represents both modes for the t-time Petri net of figure 1.

The objective of the A mode is different from the one presented here. In consequence, it differentiates the states for which there is a conflict between two transitions from the states where only one of both are fireable. Let us consider classes \mathcal{C}_{15} et \mathcal{C}_{24} (fig. 6 and table 1). Transition t_5 can be fired from the states of \mathcal{C}_{15} (whatever t_4 can or cannot be fired). But \mathcal{C}_{24} (the restricted class of \mathcal{C}_{15}) has been defined in order to characterize the temporal constraint that must be verified to fire t_4 and so t_5 does not appear as an output of this node. Using

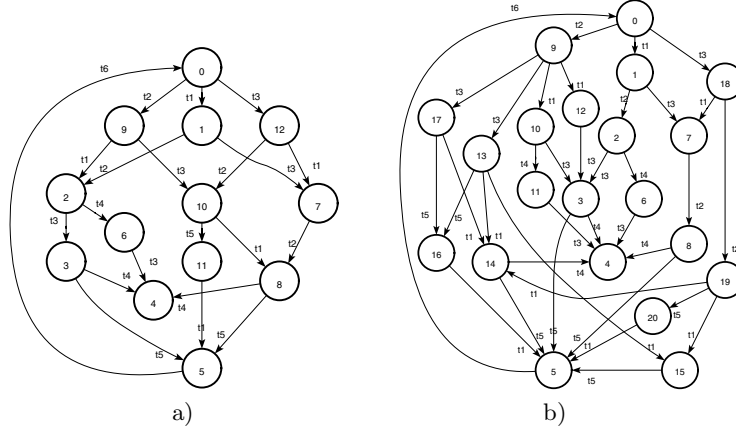


Fig. 10. Graph of classes in Tina : a) linear (W), b) atomic (A)

A mode¹, Class A14 in fig. 10.b gathers up the states where there is a conflict between t_4 et t_5 , and A15 the ones where only t_5 can be fired.

Another difference appears in the way the past is memorized. For example, C_{20} and C_{22} correspond² to a same class W5 in fig. 10.a. The corresponding temporal constraint networks $Nc_{20} = x_5$ and $Nc_{22} = x_1$ allow to preserve the name of the transition which has enabled t_6 . The memory of the past can go back beyond the last event. For example C_{13} and C_{10} are different even if the last transition fired is t_3 . Besides x_3 , Nc_{13} has variable x_1 (t_1 has enabled t_4) and Nc_{10} has variable x_2 (t_2 has enabled t_4). But in mode W and also A, there is only one class (W3 and A3 respectively).

[Yo 98] is closer to our approach. There are two differences. In our approach, when a restricted class C_r is created, the initial class C is conserved instead of replacing it for a class that is complementary to C_r . Another difference is that in our approach a class does not keep all the constraints in the past, but only the ones that are necessary to characterize it, as proved in section 3.4.

6 Conclusion

The presented approach presents a graph of classes that allows obtaining the exact temporal constraints that have to be verified by each transition firing with respect to a given firing sequence. A state class in the graph is defined by a marking and a temporal network; an arc between two classes is labeled by a

¹ Correspondence between A and C: $W_2 = (A_2, A_{10}, A_{12})$, $W_6 = (A_6, A_{11})$, $W_8 = (A_8, A_{14}, A_{15})$, $W_{10} = (A_{13}, A_{17}, A_{19})$, $W_{11} = (A_{16}, A_{20})$, $W_{12} = A_{18}$. For $i = 0, 1, 3, 4, 5, 7, 9$, $W_i = A_i$.

² Correspondence between W and C mode: $W_2 = (C_6, C_{23}, C_4)$; $W_3 = (C_{13}, C_{10})$; $W_4 = (C_{19}, C_{21})$; $W_5 = (C_{20}, C_{22})$; $W_6 = (C_{14}, C_{11})$; $W_7 = (C_5, C_3)$; $W_8 = (C_{15}, C_{24}, C_{12}, C_{17}, C_{25})$; $W_{10} = (C_7, C_9)$; $W_{11} = (C_{16}, C_{18})$; $W_0 = C_0$; $W_1 = C_1$; $W_9 = C_2$; $W_{12} = C_3$.

temporal network Nt_i delimiting the firing date of a transition t_i . The temporal constraints verified by a firing sequence are obtained by the union of the temporal constraints Nt_i attached to each arc along the corresponding path on the class graph. This set of constraints can only be derived after some transformations and calculations in the case of the other graphs of classes in the literature (it was not their objective). It is important to underline another point: the very knowledge of the temporal network associated with a firing sequence is not necessary to answer whether or not a property is verified, but it is absolutely necessary to help the designer to adjust the parameter values such that the property be verified.

Further research should consider the following issues: i) translate this approach to the p-time Petri nets, ii) extend the graph of classes to deal with fuzzy time Petri nets, that associate with a transition a fuzzy interval of firing, allowing to evaluate a possibility and necessity degree of transition firing.

References

- [Be 91] B. Berthomieu, M. Diaz : Modeling and verification of time dependent systems using Time Petri nets *IEEE Trans. on Software Engineering*, Vol 17, No 3 p.259-273 1991.
- [Be 04] B. Berthomieu, P.O. Ribet, F. Vernadat : The tool TINA: construction of abstract state spaces for Petri nets and time Petri nets, *IJPR*, Vol.42, N°14, pp.2741-2756, 15 Juillet 2004.
- [Ca 05] J. Cardoso, S. Cousy, G.Juanole : Extending time Petri nets to fuzzy time Petri nets: definition of the graph of fuzzy state class, 16th IFAC World Congress, Juillet 2005, Prague.
- [De 91] R. Dechter, I.Meiri, J. Pearl : Temporal constraint networks, *Artificial Intelligence*, vol 49, p.61-91, 1991.
- [Gh 04] M. Ghallab, D. Nau, P. Traverso : Automated Planning Theory and practice, Morgan Kaufman, 2004 ISBN 1-55860-856-7.
- [Me 85] M. Menasche : PAREDE: an automated tool for the analysis of time Petri nets, International workshop on timed Petri nets Torino July 1985, p. 162-169
- [PR 99] B. Pradin-Chézalviel, R. Valette, L.A. Künzle, Scenario duration characterization of t-timed Petri nets using linear logic, PNPM'99, 8th Int. Workshop on Petri Nets and Performance Models, Zaragoza, Spain, pp.208-217, Sep. 6-10, 1999.
- [Ri 01] N. Rivière, B. Pradin-Chézalviel, R. Valette : Reachability and temporal conflicts in t-time Petri nets, 9th Int. Workshop on Petri Nets and Performance Models, IEEE PNPM'01, Aachen, Allemagne, pp.229-238, 11-14 Sep 2001.
- [Ri 05] N. Rivière, H. Demmou, R. Valette, M. Medjoudj, Symbolic temporal constraint analysis, an approach for verifying hybrid systems, 16th IFAC, Prague, July 2005.
- [Sc 04] V. Schastai, E.A. Lima, L.A. Knzle : Sequence analysis for time Petri nets, IFAC 7th Int. Workshop on Discrete Event Systems WODES'04, France.
- [Yo 98] T. Yoneda, H. Ryuba, CTL Model checking of time Petri nets using geometric regions, *IEICE Trans. inf. & Syst.*, Vol E81-D, No. 3, pp.297-396, 1998.