Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 3, 377–394
DOI: 10.61822/amcs-2026-0024

EXTENDED TIME PETRI NETS

MARCIN RADOM[a,b,*], PIOTR FORMANOWICZ[a]

[a] Institute of Computing Science
Poznan University of Technology
ul. Piotrowo 2, 61-138 Poznań, Poland
e-mail: {Marcin.Radom,Piotr.Formanowicz}@cs.put.poznan.pl

[b] Institute of Bioorganic Chemistry
Polish Academy of Sciences
ul. Noskowskiego 12/14, 61-704 Poznań, Poland

In many complex systems that can be modeled using Petri nets, time can be a very important factor which should be taken into account during creation and analysis of the model. Time data can describe the starting moments of some actions or their duration before their immediate effects start to influence some other areas of the modeled system. Places in a Petri net often portray static components of the system, but they can also describe states. Such a state can have time restrictions, for example, indicating how long it can influence other elements in the model. Time values describing some systems may be inconsistent or incomplete, which can cause problems during the creation of the model. In this paper, a new extension of time Petri nets is proposed which allows the creation of models with different types of time data that previously were able to be properly used in separate types of well-known time Petri nets. The new time Petri net introduced here solves this problem by integrating different aspects of already existing time Petri nets into one unified structure.

Keywords: Petri nets, time constraints, tokens with lifetimes.

1. Introduction

Petri nets are one of the popular tools used to model complex systems. They can be employed for systems characterized by a large number of elements and a high number of interactions between them. Since the introduction of Petri nets (Petri, 1962), many modifications and extensions have been proposed. At a high level of classification, one can distinguish, among others, time, stochastic, continuous, hybrid, and colored Petri nets, as well as many further extensions. Recent research shows that the development of Petri net extensions is still very active, especially in directions related to time semantics, uncertainty handling or data-driven model construction. In particular, generalized stochastic Petri nets are still being developed, for example, in process-oriented settings, where such models are created from event data (van der Aalst, 2025). Moreover, probabilistic and stochastic timed extensions are used to represent changes in duration or branching behavior in real applications, including healthcare workflows (Moigne *et al*., 2025; Zhou *et al*., 2019). Other research concerns fuzzy Petri nets, which address imprecision in modeled knowledge. This direction is represented both by the recent survey work of Jiang *et al*. (2022) and by proposals that combine fuzziness with stochastic and timed semantics in a single formalism (Sobrino *et al*., 2021). Although these classes of nets are different from the formalism proposed in this paper, they show that the current research on Petri nets increasingly focuses on enriching the modeling power with respect to time, uncertainty and handling incomplete information.

Time Petri nets (TPNs) were initially introduced by Merlin (1974). The first algorithmic approach to their analysis was proposed by Berthomieu and Menasche (1983). In that paper, the so-called state classes were defined and used to analyze the states of the net using state reachability graphs. This approach was further improved by Berthomieu and Diaz (1991). In this type of Petri nets, for each transition, a separate time interval is assigned. Each transition can fire only after a certain amount of time has passed since the moment of its activation. Active transitions in a TPN must fire no later than the time

*Corresponding author

defined as the right end of the interval. In the same year as time Petri nets, the concept of timed Petri nets, also called (time) duration Petri nets (DPNs), was proposed (Ramchandani, 1974). Further analysis of this type of nets was presented by Sifakis (1980). In a DPN, a certain time is associated with each transition. In this type of time nets, each active transition is immediately fired, and tokens from its pre-places are taken at that moment. However, the production of new tokens occurs only after some time has passed, specified by the duration value assigned to the transition. An extension of this type of nets has been proposed (later called the interval-time Petri net, ITPN), where a fixed duration is replaced by a range (Popova-Zeugmann and Pelz, 2021). A separate category of time Petri nets includes those in which values specifying time are assigned to places. These nets are called Petri nets with time windows (Popova-Zeugmann, 1993). In general, each token that appears in a given place has its own lifetime. If a place is described by a certain range, it is assumed that a token cannot stay in that place longer than the time specified by the upper value of this range. The lower value specifies the minimum amount of time that must elapse for a given token before it can be used to activate any transition. These three types of Petri nets with times are well known and widely used, and new extensions and proposals for their analysis are still being proposed (e.g., Zilio *et al*., 2014; Bourdil *et al*., 2016).

Also, the current research on timed Petri nets is not limited only to reachability analysis but also includes controller synthesis, deadlock prevention, symbolic verification and new stochastic timed variants. For example, state-class based controller synthesis for a TPN has recently been revisited (Leclercq *et al*., 2023) and time-based deadlock prevention studied (Boucheneb *et al*., 2022). New probabilistic and stochastic developments also appear in related timed-net formalisms, for example, in statistical model checking for stochastic timed-arc Petri nets (Dubois *et al*., 2025). These developments confirm that the extension of Petri nets with time information remains an active research area. In this context, the proposed xTPN formalism should be viewed as a deterministic interval-based extension aimed at combining several known time-assignment mechanisms within one unified state semantics, rather than as a stochastic or fuzzy variant.

Time is obviously an important factor in the modeling and analysis of complex systems, for example, biological ones. In such a system, new compounds have to form and mature, whereas the chemical reactions that are responsible for their creation are usually not instantaneous. One of the problems that can be encountered when modeling such a system using previously described types of time Petri nets is that the time data (ranges or discrete values) can be incomplete or, depending on the source, even contradictory. This may require additional transformations and changes for the time values obtained (Yoshimasa *et al*., 2010; Mizuta *et al*., 2017). A new extension of time Petri nets, with the features of the well-known time nets, would allow much more precise model development and adaptation for complex biological systems with many different time variables.

In this paper, a new type of Petri nets, called an extended time Petri net (xTPN), is proposed. It integrates three separate types of time nets that have been described, i.e., a TPN with time intervals for an activation of transitions, an ITPN, which allows one to specify the duration of token production by transitions, and time nets, in which the time ranges are associated with places. The use of three numerical time intervals, that is, two for transitions and one for places, can be convenient because it will allow both approximate and exact values of time to be included in the model. A brief introductory definition of a net of this type with two basic propositions for token analysis algorithms based on net time variables was given in our earlier work (Radom and Formanowicz, 2022). However, in that paper, only some specific features of xTPNs were presented, which could be potentially interesting from a biological modeling point of view. In the present paper, we provide a full notation, definitions and semantics of the net. At the present stage, the proposed net is positioned primarily as a modeling and simulation formalism for systems in which several types of timing constraints must be represented simultaneously. The definitions introduced in this paper provide a formal basis for future verification-oriented research; however, the present work does not provide a complete state-space verification method or a full model-checking procedure for the xTPN formalism.

In the next sections, state changes and other interesting features, such as transformations, are given. It is worth mentioning that extended time Petri nets can be created and analyzed using a recent version of the software called Holmes (Szawulak *et al*., 2022), which is a program for building and analysis of different models using various types of Petri nets: classical, time, functional, etc.

The structure of this paper is as follows. In the next section, definitions of more basic time Petri nets than xTPNs are given, and then the proposed type of nets is formally defined. The so-called p-state and t-state are introduced, and it is explained how these two sub-states together create a full state of an extended time Petri net. The activation of transitions depends on the existence of the so-called *activating subsets of tokens*, which strongly rely on the time values associated with the net elements. This is crucial for understanding how the proposed net behaves and is explained in detail. Section 3 covers an example biochemical model of a system, presenting new possibilities offered by the proposed nets. In Section 4,

a possible transformation of the xTPNs elements (both places and transitions) is described. A summary of the new time Petri net proposal is included in the closing Section 5.

2. Extended time Petri nets

Before we present definitions of extended time Petri nets, an introduction to simpler nets, particularly those with time constraints, is given in this section.

2.1. Petri nets with time. The basic notation used is as follows: $\mathbb{N}$, $\mathbb{Q}$ and $\mathbb{R}$ denote sets of natural, rational and real numbers, respectively, $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ stands for the set of natural numbers without 0, while $\mathbb{Q}_0^+$ denotes the set of non-negative rational numbers. On the other hand, $\mathcal{N}$, $\mathcal{T}$, $\mathcal{W}$, $\mathcal{S}$ and $\mathcal{Z}$ are various tuples representing different Petri nets.

A classical Petri net is the backbone of any proposed time extension. In the literature, it is sometimes called a *skeleton* of some other type of Petri nets. It can also be referred to as the place-transition Petri net. A classical Petri net is described in the following Definition 1.

Definition 1. (*Petri net*) The Petri net is an ordered 5-tuple $\mathcal{N} = (P, T, F, V, m_0)$, where P and T are finite, nonempty and disjoint sets of places and transitions, respectively, $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs, $V : F \to \mathbb{N}^+$ is a function that assigns weights to the arcs, and $m_0 : P \to \mathbb{N}$ is an initial marking (Murata, 1989).

The structure of a Petri net is a bipartite directed graph, with vertices being places and transitions that are connected by arcs, which have weights that are positive natural numbers. A state (also called marking) of a net is a vector that contains information about the number of tokens at each place at a given moment.

We will now define *pre-set* and *post-set* notation for transitions and places. For transition t_j, its pre-set is the set of places connected with t_j by arcs directed to t_j from such places, and its post-set is the set of places connected with t_j by arcs directed from t_j to such places. The pre-set of t_j is denoted by ${}^\bullet t_j$, whereas its post-set is denoted by $t_j^\bullet$. Similarly, for a given place p_i, its pre-set is the set of transitions connected with p_i by arcs directed to p_i from such transitions, and its post-set is the set of transitions connected with p_i by arcs directed from p_i to such transitions. These sets are denoted by ${}^\bullet p_i$ and $p_i^\bullet$, respectively. More precisely, we define the following four sets:

1. ${}^\bullet t = \{p \in P : (p, t) \in F\}$ is the pre-set of transition t,
2. $t^\bullet = \{p \in P : (t, p) \in F\}$ is the post-set of transition t,
3. ${}^\bullet p = \{t \in T : (t, p) \in F\}$ is the pre-set of place p,
4. $p^\bullet = \{t \in T : (p, t) \in F\}$ is the post-set of place p.

It should be noted that the net proposed in this paper will use a modified definition of the transition pre-set, as explained in detail in Section 2.3.

Transition t in a Petri net becomes active if in each place $p_i \in {}^\bullet t$ there are at least as many tokens as the value of weight $V(p_i, t)$. An active transition can fire, i.e., it produces tokens in its post-places, yet it is not always assumed that it must fire immediately when it becomes active. However, if a transition fires, it produces tokens in each place $p_j \in t^\bullet$ in quantities equal to weights $V(t, p_j)$, while at the same time it takes tokens from each $p_i \in {}^\bullet t$ in quantities equal to weights $V(p_i, t)$.

There are many variants of Petri nets with the ability to handle time values assigned to transitions. Two especially well-known nets were introduced by Merlin (1974) (time Petri net, TPN) and by Ramchandani (1974) (timed Petri net, also known as duration-time Petri net, DPN). In Definition 2 the former is described.

Definition 2. (*Time Petri net*) A time Petri net (TPN) is a pair $\mathcal{T} = (\mathcal{N}, I)$ such that $\mathcal{N}$ is a place-transition Petri net, $I : T \to \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{+\infty\})$, and for each $t \in T$, with $I(t) = (I_1(t), I_2(t))$, it holds that $I_1(t) \leq I_2(t)$ (Popova-Zeugmann, 2013).

The classical Petri net denoted here as 5-tuple $\mathcal{N}$ is called a *skeleton* of $\mathcal{T}$. Here I is an interval function, for each transition t, assigning values $I_1(t)$ and $I_2(t)$ that are rational numbers, which are called the earliest and the latest firing times of t, respectively. An active transition will fire only after some time τ_t^{act} for which it holds that $I_1(t) \leq \tau_t^{act} \leq I_2(t)$.

The timed Petri net (DPN) has a different time function, assigning only a single rational number for each transition. This value describes the deterministic production time for a given transition. There exists an extension of a DPN, called an interval-timed Petri net (Popova-Zeugmann and Pelz, 2021; Popova-Zeugmann, 2013). Instead of a single deterministic value, a range is assigned for each transition, similarly as in a TPN. This kind of net is formally described in Definition 3.

Definition 3. (*Interval-timed Petri net*) The interval timed Petri net (ITPN) is an ordered pair $\mathcal{W} = (\mathcal{N}, D)$ such that $\mathcal{N}$ is a place-transition Petri net and $D : T \to \mathbb{Q}_0^+ \times \mathbb{Q}_0^+$ with $D(t) = (D_1(t), D_2(t))$ implies $D_1(t) \leq D_2(t)$ for every $t \in T$ (Popova-Zeugmann, 2013).

One can see that both definitions are almost identical. The real difference lies in the interpretation of functions I and D in TPNs and ITPNs, respectively. In the case of the latter, we call $D_1(t)$ the *earliest firing duration* and $D_2(t)$ the *latest firing duration*. However, the assumed changes

in this net behavior are more significant than can be seen in its definition. In a TPN, an active transition remains in an activation state for time τ_t^{act}, and then it fires. Firing itself remains an immediate event that, at the same time, takes the necessary tokens from all places from ${}^\bullet t$ and produces new ones in places belonging to $t^\bullet$. However, there are important changes. First, token production takes time, defined by the time within the range $[D_1(t), D_2(t)]$. Secondly, it is assumed that each active transition must fire immediately and take the required number of tokens from its pre-set. This behavior is defined as a rule called the *maximal step*.

When considering $\mathcal{W}$ from Definition 3, firing works as follows. Immediately when the transition becomes active, it starts its firing event. At that moment, tokens are taken from all places of this transition pre-set. Then, an internal timer for this transition starts counting the time toward some specific time τ_t^{prod} for which $D_1(t) \leq \tau_t^{prod} \leq D_2(t)$. Exactly after time τ_t^{prod}, transition t produces tokens in all places belonging to its post-set.

The last important extension of the Petri net is a type of time nets, but in this case time constraints are assigned to places. The following Definition 4 describes the P-time Petri net.

Definition 4. (*P-time Petri net*) A P-time Petri net (P-TPN) is a pair $\mathcal{S} = (\mathcal{N}, J_P)$, where $\mathcal{N}$ is a place-transition Petri net, $J_P : P \to \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\})$ assigns to each place $p \in P$ a residence-time interval $J_P(p) = [a_p, b_p]$, where $a_p \in \mathbb{Q}_0^+$ is the minimum time a token must spend in place p before it can take part in a transition firing, and $b_p \in \mathbb{Q}_0^+ \cup \{\infty\}$ is the maximum time after which the token is removed from place p (Khansa *et al*., 1996).

Each token is denoted by its age κ measured from the moment it was created in its current place. Transition t is active if, for every place $p \in {}^\bullet t$, the number of tokens in p whose age satisfies $b_p \geq \kappa \geq a_p$ is equal to or greater than $V(p,t)$ (if $b_p = \infty$, it is assumed that a token can remain in p without an upper time limit).

Firing transition t removes from each place $p \in {}^\bullet t$ exactly $V(p,t)$ tokens for which it is true that $\kappa \geq a_p$. Then, transition adds to every place $p' \in t^\bullet$ exactly $V(t,p')$ new tokens with the age equal to 0. Tokens whose age exceeds b_p are removed.

In this paper, a new type of time Petri nets is proposed which has some properties of the described nets $\mathcal{T}$ and $\mathcal{W}$, and also uses the idea of time associated with places (net $\mathcal{S}$). The extended time Petri net proposed in this paper has its own notation for such places, while the general idea was proposed by Popova-Zeugmann (1993) and later analyzed by Wegener and Popova-Zeugmann (2009). Nets with time associated with places are an extension of the nets introduced by Coolahan and Roussopoulos (1983). It is important to mention one other extension of timed Petri nets. In the work of Bolognesi *et al*. (1990), the so-called timed-arc Petri net was proposed. In such nets, a time interval is associated with arcs, which then allows tokens to pass only if their lifetime is within the range. Some properties of such nets were analyzed by Jacobsen *et al*. (2011).

2.2. Definitions of extended time Petri nets. In this section, the basic definitions of the proposed net are given. The extended time Petri net without a state is described in Definition 5.

Definition 5. (*Extended time Petri net without state*) An extended time Petri net without state is an ordered 3-tuple $\mathcal{Z} = (\mathcal{N}, I, J)$, where $\mathcal{N} = (P, T, F, V)$ is a place-transition Petri net, $I : T \to \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\}) \times \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\})$, and for every $t \in T$ where $I(t) = (I_1(t), I_2(t), I_3(t), I_4(t))$ it holds that $I_1(t) \leq I_2(t)$, and $I_3(t) \leq I_4(t)$, $J : P \to \mathbb{Q}_0^+ \times (\mathbb{Q}^+ \cup \{\infty\})$, and for every $p \in P$ where $J(p) = (J_1(p), J_2(p))$ it holds that $J_1(p) < J_2(p)$.

Function I assigns to each transition four ordered values $I_1(t), I_2(t), I_3(t), I_4(t)$, which are in fact two pairs defining some specific time intervals. The first pair $(I_1(t), I_2(t))$ specifies a time interval within which an active transition must be fired. Later, for simplicity, these will be referred to as α values, where $I_1(t) = \alpha_t^L$ and $I_2(t) = \alpha_t^U$. The letters L and U stand for the lower and upper bounds of the interval, respectively. These two values always satisfy the inequality $\alpha_t^L \leq \alpha_t^U$ for any transition $t \in T$.

The second pair of values $I_3(t)$ and $I_4(t)$ specifies the minimum and maximum token production time for a transition that has been fired. They will be later referred to as β values: $I_3(t) = \beta_t^L$ and $I_4(t) = \beta_t^U$ for each transition $t \in T$.

Transitions from net $\mathcal{Z}$ start to produce tokens after some time τ_t^α, for which $0 \leq \alpha_t^L \leq \tau_t^\alpha \leq \alpha_t^U$. When production starts, the transition immediately takes tokens from every place $p \in {}^\bullet t$, but the production of new tokens ends after time τ_t^β for which $0 \leq \beta_t^L \leq \tau_t^\beta \leq \beta_t^U$. It should be noted that over time τ_t^α a transition must remain active, but when production has started the conditions for transition activation are not checked for the duration of τ_t^β. Therefore, the production phase cannot be stopped once it has started.

Each transition, at any moment, is in one of three states: inactive, active, or producing tokens. With each transition, two timers are associated. The first one, later referred to as u_t, is the activation timer. When production starts, tokens are taken from places that are elements of ${}^\bullet t$ and the second timer w_t starts counting. The inactive transition has its timers set to a special symbol $\#$. A more detailed description will be given in the section describing the so-called t-state of $\mathcal{Z}$ (Definition 21).

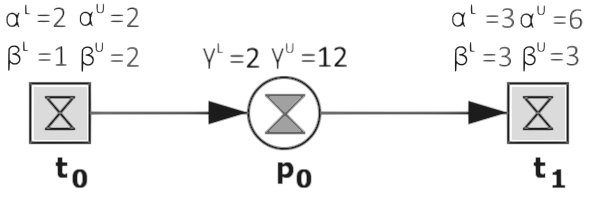


Fig. 1. Extended time Petri net with time intervals α and β assigned to transitions and time interval γ assigned to a place.

Function J assigns to each place $p \in P$ two values, $J_1(p)$ and $J_2(p)$. The lower one denotes the minimum maturity time for a token. Tokens with an age lower than this value cannot activate the transition, nor can they be consumed during the production event. The second value specifies the maximum time that a token can stay in that place. They will be referred to as values γ such that $J_1(p) = \gamma_p^L$ and $J_2(p) = \gamma_p^U$. Although γ_p^L can be zero, γ_p^U must be a positive number (which means that the maximum token lifetime allowed in a place must be longer than 0). The value of γ_p^L specifies the time that must pass for each token before it can activate the post-set transitions of a place where it resides. The maximum lifetime of a token in a place is given as γ_p^U, and up to that moment the token can be consumed by some transition belonging to a pre-set of such a place, after which the token is removed.

Remark 1. (*Net structure and state*) Similarly to the *skeleton* terminology used for timed Petri nets, Definition 5 specifies only the static structure of an xTPN, i.e., the underlying place-transition Petri net (P, T, F, V) together with the time-interval assignment functions I (transition timing) and J (place timing). The dynamic state used by the operational semantics is introduced later in Section 2.5 as $z = (m, h)$ (Definition 22), where m is the p-state (multisets of tokens in places) and h is the t-state (activation and production timers of each transition).

An example net is presented in Fig. 1. It should be noted that both transitions in the example, t_0 and t_1, can have two intervals assigned, even though, e.g., t_0 will not consume any tokens when it fires, nor will t_1 produce ones. The production time for t_1 is deterministic because both β values are equal. The same is true for an activation time of t_0, which shows the flexibility of the proposed nets elements in handling time data.

2.3. Types of arcs in an xTPN. In the proposed Petri net, two special kinds of arcs are available, namely, read arcs and inhibitors. The tokens are not moved through these arcs during the production phase. A read arc, with respect to activation, behaves like any other arc directed from a place to a transition. However, no tokens are consumed or produced through it. An inhibitor arc ensures that, if a place contains at least as many mature tokens as its weight, the transition to which it is connected cannot be activated, regardless of the number of tokens in places belonging to its pre-set.

Within set of arcs $F \subseteq (P \times T) \cup (T \times P)$, there are four disjoint subsets: $F^{in} \subseteq (P \times T)$, $F^{out} \subseteq (T \times P)$, $F^{inh} \subseteq (P \times T)$, $F^{read} \subseteq (P \times T) \cup (T \times P)$, such that

- $F = F^{in} \cup F^{out} \cup F^{inh} \cup F^{read}$,
- $F^{in} \subseteq (P \times T) \setminus F^{inh} \setminus F^{read}$,
- $F^{out} \subseteq (T \times P) \setminus F^{read}$.

These four subsets of F are disjoint. Additionally, $F^{read} \subseteq F$ and $F^{read} \subseteq (P \times T) \cup (T \times P)$. The read arcs do not cause token flow and form a set F^{read} such that $\forall (p,t) \in P \times T : (p,t) \in F^{read} \iff (t,p) \in F^{read}$, and both weights are equal: $V(p,t) = V(t,p)$.

As for the inhibitor arcs, they will be discussed later in detail, but it is assumed that the direction of such an arc is towards a transition, from some place. When there are enough tokens (with respect to arc weight and their lifetimes) in a place connected with the inhibitor arc to a transition, such an arc will completely block transition activation. For the classical Petri net, sets of pre- and post-sets have already been defined. However, for the xTPN, the definition of a pre-set of a transition must take into account various types of arcs, that is, various disjoint subsets of F. This is explained in Definition 6.

Definition 6. (*Pre-sets and post-set of transitions in an xTPN*) For each transition $t \in T$ its places directly connected with t can be assigned to four different subsets:

1. $t^\bullet = \{p \in P : (t,p) \in F^{out}\}$ is a subset of places from a post-set of t connected with a normal arc,
2. $^\bullet t = \{p \in P : (p,t) \in F^{in}\}$ is a subset of places from a pre-set of t connected with a normal arc,
3. $^\bullet t^I = \{p \in P : (p,t) \in F^{inh}\}$ is a subset of places connected by an inhibitor arc with transition t,
4. $^\bullet t^R = \{p \in P : (p,t) \in F^{read}\}$ is a subset of places connected by a read arc with transition t.

2.4. Multisets of type *K* and *M*. Since tokens are separate objects with individual lifetimes, a special type of multisets must be defined to store values representing such objects. A special class K of multisets is given in Definition 7. Such multisets are associated with places and contain values that represent tokens of a given place.

At this point, an important remark is necessary regarding the symbols of inclusion and set difference. From now on, we will work almost exclusively with *multisets* (with multiplicities), typically denoted by K and M.

Throughout this paper, the symbol $\subseteq$ means *multiset inclusion* (with respect to multiplicities) while the symbol $\setminus$ stands for *multiset difference*.

Definition 7. (*Multiset of type K*) Let ${}^{a}_{b}K$, where $a > 0$, $a \geq b \geq 0$, $a \in \mathbb{Q}^{+} \cup \{\infty\}$ and $b \in \mathbb{Q}^{+}_{0}$, be a multiset of non-negative real numbers less than or equal to some positive rational value a. The elements of this multiset are numbers from the closed interval $[b, a]$, which can be written as follows: $\kappa_i \in {}^{a}_{b}K$, where $\kappa_i \in \mathbb{R}^{+}_{0}$ and $0 \leq b \leq \kappa_i \leq a$. In the special case, ${}^{a}_{b}K = \{\#\}$.

In the above definition, a is an admissible maximum value of the elements of ${}^{a}_{b}K$ and b is the minimum allowed value of the elements of multiset ${}^{a}_{b}K$. Sometimes, when the values a and b are not important, they will be omitted for the sake of clarity. The multisets ${}^{a}_{b}K$ will be assigned to the places or will describe in some other way the distribution of tokens in $\mathcal{Z}$. For example, multisets K assigned to places to describe their tokens will always have the value $b = 0$. On the other hand, for example, the so-called activating subsets of tokens (which are also multisets of type K) will usually have a non-zero value b. By general convention, the symbol or letter on the upper right of the multiset type K will always inform us about the specific functions of that multiset. A multiset of this kind can be empty. The case where $K = \{\#\}$ will be employed later, for example, in definitions dealing with transition activation constraints. For most of the definitions in this paper, the letter K will be used for various multisets of this kind. Sometimes, however, some other letters will be used to denote multisets *of type* K.

After assigning multiset K to a specific place, the name of that place will be given as the bottom right index of multiset K, for example, as in ${}^{a}_{b}K_p$. Since positive non-zero value γ^{U}_{p} is associated with some specific place p, assigning a multiset ${}^{a}_{b}K$ to a particular place p will precisely set the values a and b. The value a will always be equal to γ^{U}_{p}. Depending on the purpose of the multiset, the two most common values of b are zero or γ^{L}_{p}.

In Definition 8, a multiset of type M is described. This special kind of multiset will be used to store multisets of type K, and the most common use of it will be to describe the sub-state of an xTPN concerning places.

Definition 8. (*Multiset type M*) Let M be a multiset whose elements are multisets K. Therefore, $M = \{{}^{a_1}_{b_1}K_1, {}^{a_2}_{b_2}K_2, ..., {}^{a_n}_{b_n}K_n\}$. For a given net $\mathcal{Z}$, it is always true that $|M| = |P|$.

Multisets of this type generally describe the so-called p-state of the net, which contains data about the lifetimes of all tokens that exist at places in a given moment. The multisets M will have some specific subtypes defined later but will always satisfy $|M| = |P|$. In particular, the following operations on such multisets should be precisely defined:

- The elapsed time increases the values of elements in multisets K, which describe the lifetime of the tokens. This increase is always the same for the entire net. Related to this is also the problem of removing elements from multiset K that, due to the elapsed time, no longer satisfy the condition of belonging to K. The same amount of time is also added to the transition timers u_t and w_t described further on.
- Creating new tokens in place p_j, that is, adding new elements to the corresponding multiset K_{p_j}. This corresponds to the end of a production phase of some transitions.
- Checking whether multiset K_{p_j} contains a subset with certain properties to determine whether a transition is active or inactive.
- Removal of a subset of tokens from a place, that is, some elements of K_{p_j}, as a result of starting a transition production phase.

Definition 9 shows how the values of all elements of multisets K are modified over time.

Definition 9. (*Elapsing of time in multisets K and M*) Let ${}^{a_j}_{0}K_{p_j}$ be a non-empty multiset specifying the current lifetime of each token in place p_j, $a_j = \gamma^{U}_{p_j}$. The elapsed time τ modifies all elements of ${}^{a_j}_{0}K_{p_j}$, thus creating multiset ${}^{a_j}_{0}K'_{p_j}$ which is denoted by ${}^{a_j}_{0}K_{p_j} \xrightarrow{\tau} {}^{a_j}_{0}K'_{p_j}$. Multiset ${}^{a_j}_{0}K'_{p_j}$ is defined as follows: ${}^{a_j}_{0}K'_{p_j} = \{\kappa'_x = \kappa_x + \tau \mid \kappa_x \in {}^{a_j}_{0}K_{p_j} \wedge \kappa'_x \leq a_j\}$. The elapsed time for multiset $M = \{{}^{a_1}_{0}K_{p_1}, {}^{a_2}_{0}K_{p_2}, \ldots, {}^{a_n}_{0}K_{p_n}\}$ changes multiset M into M', which is denoted by $M \xrightarrow{\tau} M'$, where $M' = \{{}^{a_1}_{0}K'_{p_1}, {}^{a_2}_{0}K'_{p_2}, \ldots, {}^{a_n}_{0}K'_{p_n}\}$.

Obviously, if $K_{p_j} = \emptyset$, then for any τ we have $K_{p_j} \xrightarrow{\tau} K'_{p_j}$ with $K'_{p_j} = \emptyset$. New elements can be added to and removed from a multiset of type K. Definition 10 specifies how to add new elements to multiset K_j.

Definition 10. (*Addition of elements to multisets K*) Let K_{p_j} be a multiset assigned to place p_j, and let $v \in \mathbb{N}$. Adding v new elements to multiset K_{p_j} creates a new multiset K'_{p_j} which is denoted by $K'_{p_j} = K_{p_j} + v = K_{p_j} \uplus K^{v}$, where $K^{v} = \{\kappa_i = 0 : i = 1, \ldots, v\}$.

Since all sets are multisets, each zero in the auxiliary multiset K^{v} defines a new distinct token created at some point in time. For example, let $K_{p_j} = \{0, 0, 1, 2, 3, 3\}$ and $v = 3$. Then the operation $K_{p_j} + 3$ creates a new multiset $K'_{p_j} = \{0, 0, 0, 0, 0, 1, 2, 3, 3\}$. It should be noted that the parameters a_j and b_j, which are usually placed on the left side of multisets K, have not been given, because their values are irrelevant in this operation.

The next operation that can be performed on multiset K_p determines how the elements of the set are removed,

which represents taking tokens from each place $p \in {}^\bullet t$, when some active transition t begins its production phase. However, it is first necessary to define a certain very important multiset of type K, which will be called an *activating subset of tokens*. This multiset is specified by Definition 11.

Definition 11. (*Activating subset of tokens*) Let ${}^{a_j}_{0}K_{p_j}$ be a multiset of tokens of place p_j, $a_j = \gamma^U_{p_j}$, and let $v \in \mathbb{N}$ be the weight of an arc from place p_j to transition t. The subset ${}^{a_j}_{b_j}K^{sub}_{p_j-t} \subseteq {}^{a_j}_{0}K_{p_j}$, where $b_j = \gamma^L_{p_j}$, is defined as ${}^{a_j}_{b_j}K^{sub}_{p_j-t} = \{\kappa_x \in {}^{a_j}_{0}K_{p_j} : \kappa_x \geq b_j\}$. If $|{}^{a_j}_{b_j}K^{sub}_{p_j-t}| \geq v$, it means that in ${}^{a_j}_{0}K_{p_j}$ there exists an activating subset (of tokens) denoted by ${}^{a_j}_{b_j}K^{act}_{p_j-t}$. This subset has the specific cardinality $|{}^{a_j}_{b_j}K^{act}_{p_j-t}| = v$ and contains the v largest elements of ${}^{a_j}_{b_j}K^{sub}_{p_j-t}$.

The following Definition 12 makes use of the activating subset of tokens to accurately describe the operation performed in an xTPN by each firing transition when it enters the production phase.

Definition 12. (*Removal of elements from multisets* K_p) Let ${}^{a}_{0}K_p$ be a multiset of tokens of place p, $a = \gamma^U_p$, $b = \gamma^L_p$, and let $v \in \mathbb{N}$ denote the weight of arc $V(p,t)$. Suppose that there exists a subset ${}^{a}_{b}K^{sub}_{p-t}$ such that $|{}^{a}_{b}K^{sub}_{p-t}| \geq v$. Removing v elements from multiset ${}^{a}_{0}K_p$, denoted by ${}^{a}_{0}K_p - v$, creates a new set ${}^{a_j}_{0}K'_p = {}^{a_j}_{0}K_p \setminus {}^{a}_{b}K^{act}_{p-t}$, where the activating subset is given by ${}^{a}_{b}K^{act}_{p-t} = \{\kappa_1, \kappa_2, \cdots, \kappa_v \in {}^{a}_{0}K_p : \sum_{x=1}^{v} \kappa_x \ \textit{is maximal}\}$.

The activating subset of tokens ${}^{a}_{b}K^{act}_{p-t}$ determines which elements are removed from ${}^{a}_{0}K_p$ when a given transition begins its production phase. It is always a subset of multiset ${}^{a}_{b}K^{sub}_{p-t}$ (which itself is a subset of ${}^{a}_{0}K_{p_j}$), and its cardinality is v, i.e., the number of elements to be removed when transition t is fired. This value is the weight of the arc connecting place p_j with transition t.

Note a particular feature of the activating subset ${}^{a}_{b}K^{act}_{p-t}$. Namely, according to Definition 12, this multiset consists only of the elements with the highest values being elements of ${}^{a}_{0}K_p$. This is specified by the condition of $\sum_{x=1}^{v} \kappa_x$ being maximal. In other words, the elements removed from multiset ${}^{a}_{0}K_p$ will always be the oldest tokens, represented by the elements with the largest values. Throughout this paper, this token-taking strategy will always be used, i.e., whenever a transition must draw tokens from its places belonging to ${}^\bullet t$, it takes those represented by the largest values in multiset K. This semantic maximizes the potential number of firings (i.e., production starting events), which will be discussed later.

In Definition 13 an inclusion relation of two M-type multisets is presented. Later, it will be used, for example, to determine if a given transition is active in the current state of $\mathcal{Z}$.

Definition 13. (*Inclusion of multisets of type* M) For two multisets of the same size $M = \{K_1, \ldots, K_n\}$ and $M' = \{K'_1, \ldots, K'_n\}$ we say that M' is included in M, which is formally written as $M' \leq M$ if $\underset{j=1,\ldots,n}{\forall} K'_j \subseteq K_j$. Otherwise, $M' \not\leq M$.

The following two operations define the rules for adding two multisets M (Definition 14) and subtracting two multisets M (Definition 15). These are required to define operations for adding and removing, respectively, a number of tokens in the net. The next two definitions (Definitions 17 and 18) will introduce special types of multisets representing tokens consumed and produced.

Definition 14. (*M-addition of multisets* M) Let two multisets $M_1 = \{K^1_1, K^1_2, \cdots, K^1_n\}$ and $M_2 = \{K^2_1, K^2_2, \cdots, K^2_n\}$ be given. An operation of M-addition of two multisets creates multiset $M' = M_1 + M_2$ such that $M' = \{K'_1, K'_2, \cdots, K'_n\}$, where, for each $j = 1, 2, \ldots, n$, $K'_j = K^1_j \uplus K^2_j$.

Definition 15. (*M-subtraction of multisets* M) Let two multisets $M_1 = \{K^1_1, K^1_2, \cdots, K^1_n\}$ and $M_2 = \{K^2_1, K^2_2, \cdots, K^2_n\}$ be given. An operation of M-subtracting multiset M_2 from multiset M_1 creates multiset $M' = M_1 - M_2$ such that $M' = \{K'_1, K'_2, \cdots, K'_n\}$, where, for each $j = 1, 2, \ldots, n$, $K'_j = K^1_j \setminus K^2_j$.

It should be explained why, for multisets of type M, addition and subtraction are denoted by normal plus and minus operators. This is based on the fact that such multisets contain elements that are multisets of type K. Multiset addition and subtraction for all multisets M are therefore performed on their elements. Using corresponding sets operators on the entire multisets M could be misleading, and the notation applied allows easy and accurate distinction between these two subtypes of multisets.

Definition 16 introduces the notion of support of multiset M.

Definition 16. (*Support of a multiset of type* M) Let M be a multiset such that $M = \{K_1, K_2, \cdots, K_n\}$ and each multiset K_p is assigned to place p. We define the support of M as $\mathrm{s}(M) = \{p \in P : K_p \notin \{\emptyset, \{\#\}\}\}$.

Two elementary operations in Petri nets concern taking a certain number of tokens from places belonging to ${}^\bullet t$ and creating a certain number of tokens in places belonging to $t^\bullet$. In the proposed net, two definitions of specific multisets, denoted by $M^{t\oplus}$ and $M^{t\ominus}$, will be given. These two multisets will have specific K multisets as elements, which in turn contain numbers that denote,

respectively, all new tokens created by a transition that has completed its production phase (Definition 17) and tokens that a transition takes from each place $p \in {}^\bullet t$ when it starts its production phase (Definition 18).

Definition 17. (*Multiset* $M^{t\oplus}$) Let $t \in T$. The multiset $M^{t\oplus} = \{K_1^{t\oplus}, \ldots, K_{|P|}^{t\oplus}\}$ contains multisets $K_j^{t\oplus}$ such that

$$K_{p_j}^{t\oplus} := \begin{cases} \{\kappa_x = 0 : x = 1, \ldots, v\} & \text{if } p_j \in t^\bullet, \\ \emptyset & \text{if } p_j \notin t^\bullet, \end{cases}$$

where $v = V(t, p_j)$ if $p_j \in t^\bullet$.

The multiset $M^{t\ominus}$ is introduced in Definition 18. It should be noted that its creation is conditioned by the existence of an activating subset of tokens.

Definition 18. (*Multiset* $M^{t\ominus}$) Let $t \in T$, and for every place $p_j \in \{{}^\bullet t \cup {}^\bullet t^R\}$ let there exist an activating subset of tokens $K_{p_j-t}^{act}$. Multiset $M^{t\ominus} = \{K_1^{t\ominus}, \ldots, K_{|P|}^{t\ominus}\}$ contains multisets $K_{p_j}^{t\ominus}$ such that

$$K_{p_j}^{t\ominus} := \begin{cases} K_{p_j-t}^{act} & \text{if } p_j \in {}^\bullet t, \\ \emptyset & \text{if } p_j \notin {}^\bullet t. \end{cases}$$

In Definition 17, multiset $M^{t\oplus}$ is presented. Each multiset $K_{p_j}^{t\oplus} \in M^{t\oplus}$ (if not empty) describes new tokens that will be added to specific places if some transition t completes its production phase therefore, $|K_{p_j}^{t\oplus}| = V(t, p_j)$. For places $p_j \notin {}^\bullet t$, multiset $K_{p_j}^{t\oplus} = \emptyset$.

Similarly, Definition 18 specifies multisets of tokens to be removed from the net when some transition t starts its production phase. Tokens are described by the corresponding numbers included in multisets $K_{p_j}^{t\ominus}$, for $p_j \in {}^\bullet t$. For the remaining places, these are empty sets. It should be noted that, as already specified in Definition 12, these are elements that represent the oldest tokens stored inside an activating subset of tokens. Therefore, exactly $V(p_j, t)$ largest numbers are assigned as $K_{p_j}^{t\ominus} = K_{p_j-t}^{act}$.

2.5. State of an extended time Petri net. A state of an extended time Petri net $\mathcal{Z}$ consists of two sub-states. They will be called p-state and t-state, and describe token distribution at places and time values assigned to transitions. Each token of the proposed net is a distinguishable object with a finite lifetime. Even when two tokens in one place have identical time values, they are considered to be independent, different objects in the net. Definition 20 describes the p-state of the net, while Definition 19 explains the more general p-marking.

Definition 19. (*p-Marking of extended time Petri net* $\mathcal{Z}$) Let P be a set of all places in $\mathcal{Z}$, where $n = |P|$, and let multiset $M = \{{}^{a_1}_0K_1, {}^{a_2}_0K_2, \ldots, {}^{a_n}_0K_n\}$. Then, the p-marking of $\mathcal{Z}$ is a function $y : P \to M$ such that $y(p_j) = {}^{a_j}_0K_{p_j}$.

Definition 20. (*p-State of the xTPN*) Let P be a set of places of $\mathcal{Z}$, and let $M = \{{}^{a'_1}_0K'_1, {}^{a'_2}_0K'_2, \ldots, {}^{a'_n}_0K'_n\}$, where $n = |P|$. The p-state of $\mathcal{Z}$ is a function $m : P \to M_0$ that assigns elements of multiset $M_0 = \{{}^{a_1}_0K_1, {}^{a_2}_0K_2, \ldots, {}^{a_n}_0K_n\}$ to places in such a way that $m(p_j) = {}^{a_j}_0K_{p_j}$ and ${}^{a_j}_0K_{p_j} = \{\kappa_x \in {}^{a'_j}_0K'_j \wedge \kappa_x \le \gamma_{p_j}^U\}$.

Evidently, not every p-marking is also a proper p-state. For that, it is required that each element of the assigned multiset ${}^{a_j}_0K_j$ be less than or equal to $\gamma_{p_j}^U$.

Given a p-state of the net (i.e., the distribution of tokens, their number and time values in all places), it is possible to determine whether a transition is active (denoted as t^{act}), inactive ($t^{\#}$) or producing tokens (t^{prod}). These are the three basic states of any transition in $\mathcal{Z}$. Definition 21 introduces the t-state of $\mathcal{Z}$.

Definition 21. (*t-State of* $\mathcal{Z}$) Let T be a set of all transitions of $\mathcal{Z}$. Function h describes the t-state of $\mathcal{Z}$ and is defined as $h : T \to (\mathbb{R}_0^+ \cup \{\#\}) \times (\mathbb{R}_0^+ \cup \{\#\})$. It assigns to each transition an ordered pair of elements (u_t, w_t) in such a way that at most one element of the pair can be a real number. If for a given transition $u_t \in \mathbb{R}_0^+$, then $w_t = \#$, and if $w_t \in \mathbb{R}_0^+$, then $u_t = \#$. Additionally, when an element of the pair is a real number, we have $0 \le u_t \le \alpha_t^U$ or $0 \le w_t \le \beta_t^U$. A special case is where $h(t) = (\#, \#)$.

Transition t can be in one of three t-states determined by function h, depending on the values of the elements in pair (u_t, w_t). The possible three cases are as follows:

1. $h(t) = (u_t = \#, w_t = \#)$ specifies that transition t is inactive and does not produce tokens. This state is denoted as $t^{\#}$;
2. $h(t) = (u_t \in \mathbb{R}_0^+, w_t = \#)$ specifies that transition t is currently active and does not produce tokens. This state is denoted as t^{act};
3. $h(t) = (u_t = \#, w_t \in \mathbb{R}_0^+)$ describes a transition that is currently in the production state. It should be noted that in this state activation rules (as explained later) do not apply. This state is denoted as t^{prod}.

A transition that is inactive can become active, which will be formally described later. A transition in the active state can either stop being active or start the production of tokens. It is important to discuss the third case, when a transition is in the production state. Then, its potential activation or deactivation is not considered. The activation state can be stopped before the start of the production state. However, the latter will always last for τ_t^β time units, even if the conditions for the transition to

be active are no longer satisfied because they are simply not considered during that time. When production ends, the transition changes its t-state to active or inactive.

Now the full state of the extended time Petri net can be given in Definition 22.

Definition 22. (*State of the extended time Petri net $\mathcal{Z}$*) The state of the extended time Petri net $\mathcal{Z}$ is the pair $z = (m, h)$.

Remark 2. (*State space of the xTPN*) Any specific state $z = (m, h)$ contains real values representing time (token ages in m and transition timers in h). Therefore, the set of reachable states may be infinite, as in other timed formalisms. Moreover, compared with other time Petri nets, an xTPN state contains additional components: individual token ages in places and separate activation and production timers determining discrete transition phases. As a consequence, the state space is significantly larger than in other time nets and its exhaustive formal verification by, e.g., a direct state enumeration may be feasible only for restricted or relatively smaller models.

For verification and reachability analysis it is common to group specific states into symbolic state classes consisting of a discrete part (token multiplicities and discrete transition phases) together with a set of constraints over timing variables. In an xTPN, the time variables are explicitly bounded by the model parameters: $0 \le u_t \le \alpha_t^U$, $0 \le w_t \le \beta_t^U$, and the token ages in place p satisfy $0 \le \kappa_x \le \gamma_p^U$ (Definitions 20 and 21). These bounds provide a basis for future symbolic or constraint-based analysis. However, the construction of complete symbolic state-space abstractions for the full xTPN formalism is not considered in this paper.

2.6. Transition activation. To determine whether a given transition t is active, one must look more closely at the elements of all multisets K_{p_j}, for $p_j \in {}^\bullet t$. Suppose that place p_j is the only element in ${}^\bullet t_i$ and that they are connected by an arc with a weight equal to two. For transition t_i to be active, multiset K_{p_j}, which describes tokens of place p_j, must contain at least two numbers, κ_1 and κ_2, such that $\kappa_1 \ge \gamma_{p_j}^L \wedge \kappa_2 \ge \gamma_{p_j}^L$.

To formally determine what it means for a transition to be active, an activating subset of tokens has already been defined in Definition 11. Transition activation occurs only if, for each place $p_j \in {}^\bullet t$, there exists an activating subset of tokens. In simpler words, one can say that, in order for a transition to be active, there must be a sufficient number of tokens (determined by an arc weight) in all places $p \in {}^\bullet t$. The tokens that are considered when creating that activating subset are those with a sufficiently long life in p_j (given by the value $b_j = \gamma_{p_j}^L$).

It should be noted that if transition t has many places in ${}^\bullet t$, for example, p_1 and p_2, in order for it to be active, it must have activating subsets in both K_{p_1} and K_{p_2}. To distinguish between them, the lower right indices of such activating subsets will be given in the form of a place-transition, that is, in that example there must exist ${}^{a_1}_{b_1}K^{act}_{p_1-t} \in {}^{a_1}_{0}K_{p_1}$ and ${}^{a_2}_{b_2}K^{act}_{p_2-t} \in {}^{a_2}_{0}K_{p_2}$.

The existence of an inhibitor arc requires the definition of the blocking subset of tokens, given below.

Definition 23. (*Blocking subset of tokens*) Let ${}^a_0K_p$ be a multiset of tokens of place p, $a = \gamma_p^U$, and let $b = \gamma_p^L$. For an inhibitor arc $(p, t) \in F^{inh}$ with weight $v = V(p, t)$, we define multiset

$${}^a_bK^{blk}_{p-t} = \{\kappa \in {}^a_0K_p \,:\, \kappa \ge b\}.$$

The inhibitor prevents (and stops) the activation of transition t if and only if $|{}^a_bK^{blk}_{p-t}| \ge v$.

Blocking a transition has priority over its activation, i.e., over the rules governing the existence of an activating subset of tokens. In the presence of an inhibitor arc and when $|{}^a_bK^{blk}_{p-t}| \ge v$, the consideration of activating subsets ceases to be relevant. It should also be noted that, when a transition is already in a production phase, blocking rules do not apply.

Definition 24 formally describes the conditions for transition activation.

Definition 24. (*Activation of a transition*) Let $t \in T$ be a transition, and let $n = |P|$, $m : P \to M$ describe the p-state of the net. Let $h(t) = (u \in \mathbb{R}_0^+, w = \#)$ or $h(t) = (u = \#, w = \#)$, meaning that t is active or inactive. For transition t, we determine multiset $M^t = \{K^t_{p_1}, \cdots, K^t_{p_n}\}$ called the activation multiset for t such that

$$K^t_{p_j} = \begin{cases} {}^{a_j}_{b_j}K^{act}_{p_j-t} \quad \text{if } p_j \in \{{}^\bullet t \cup {}^\bullet t^R\} \text{ and} \\ \quad |{}^{a_j}_{b_j}K^{sub}_{p_j-t}| \ge V(p_j, t), \\ \{\#\} \quad \text{if } p_j \in \{{}^\bullet t \cup {}^\bullet t^R\} \text{ and} \\ \quad |{}^{a_j}_{b_j}K^{sub}_{p_j-t}| < V(p_j, t), \\ \{\#\} \quad \text{if } p_j \in {}^\bullet t^I \text{ and} \\ \quad |{}^{a_j}_{b_j}K^{blk}_{p_j-t}| \ge V(p_j, t), \\ \emptyset \quad \text{otherwise.} \end{cases}.$$

Transition t is active in the p-state described by M if, and only if, for its activation multiset M^t it holds that $M^t \le M$.

Read arcs and inhibitor arcs do not change the number of tokens during firing and production. They only affect the activation condition. The above definition uses the notation introduced in Definitions 11 and 13. The former defines an activating subset of tokens with sufficient lifetimes. Multiset $M = \{{}^{a_1}_0K_1, {}^{a_2}_0K_2, \ldots, {}^{a_n}_0K_n\}$, where $n = |P|$, describes the tokens in all net places. If $p_j \in {}^\bullet t$ and its corresponding

multiset ${}^{a_1}_{0}K_j$ contains an activating subset of tokens, then this subset will also be an element of M^t. If the activating subset of tokens for p_j does not exist, then $K_{p_j-t} = \{\#\}$.

In the case where $K_{p_j} = \{\#\}$ for any $p_j \in {}^\bullet t$, set M^t will not be a subset of M (Definition 13), which means that transition t is inactive. Figure 2 shows an example of two transitions: active and inactive according to the rules described in Definition 24.

In the left part of Fig. 2, transition t_0 is active according to Definition 24. For place p_0, the arc weight is $V(p_0, t_0) = 3$ and the lower bound of the time interval is $\gamma^L_{p_0} = 1$, so at least three tokens with age not smaller than 1 are required. The upper bound is not important for this example, and it is set as $\gamma^U_{p_0} = 100$. Since $K_{p_0} = \{0, 1, 2, 3\}$, the tokens old enough are $\{1, 2, 3\}$, and they form the activating subset for p_0. For place p_1, the arc weight is $V(p_1, t_0) = 4$, $\gamma^L_{p_1} = 1$ and $\gamma^U_{p_1} = 100$, so four mature tokens are required. Since $K_{p_1} = \{0, 0, 1, 2, 2, 4\}$, the activating subset for p_1 is $\{1, 2, 2, 4\}$. Therefore, a multiset of activating subsets is $M^{t_0} = \{\{1, 2, 3\}, \{1, 2, 2, 4\}\}$.

Because both activating subsets are contained in the corresponding elements of $M = \{\{0, 1, 2, 3\}, \{0, 0, 1, 2, 2, 4\}\}$, we have $M^{t_0} \leq M$, and thus t_0 is active.

In the right part of Fig. 2, transition t_0 is not active. Place p_1 has the same multiset K_{p_1}, so its activating subset remains $\{1, 2, 2, 4\}$. However, for p_0 we now have $K_{p_0} = \{0, 1, 2\}$. Since $V(p_0, t_0) = 3$ and only two tokens satisfy the condition $\kappa_x \geq \gamma^L_{p_0} = 1$, no activating subset exists for p_0. Consequently, the first component of M^{t_0} is replaced by $\{\#\}$, and we obtain $M^{t_0} = \{\{\#\}, \{1, 2, 2, 4\}\}$.

Since $M = \{\{0, 1, 2\}, \{0, 0, 1, 2, 2, 4\}\}$, it follows that $M^{t_0} \not\leq M$. Therefore, t_0 is not active.

2.7. Net semantics. In an xTPN, each change in the state depends on the passage of time. Before formal definitions for state changes are given, we will summarize the basic semantics of the net.

- *Activation phase.* Transition t_i becomes active (denoted as $t^\# \rightarrow t^{act}$) if in all places from ${}^\bullet t \cup {}^\bullet t^R$ there exists an activating subset of tokens (Definition 24), and obviously there is no blocking subset if inhibitor arcs exist. Upon activation, counter u_i for transition t_i is initialized. It will count from zero towards some value within the range $[\alpha^L_i, \alpha^U_i]$. The value of u_i increases with time τ. When the transition state changes from activation to production, this change is denoted as $t^{act} \rightarrow t^{prod}$.

- *Production phase.* Transition t_i consumes tokens from all places $p \in {}^\bullet t$ (based on the arc weights). Counter w_i for transition t_i is initialized and will count from zero towards some value within the range $[\beta^L_i, \beta^U_i]$. The value of w_i increases with time τ. When the transition finally produces tokens at all places in set $t_i{}^\bullet$, it becomes active or inactive depending on the existence or not of the activation subsets (such a change is respectively denoted by $t^{prod} \rightarrow t^\#$ or $t^{prod} \rightarrow t^{act}$). A special case can be observed when the activation interval for a transition is zero. In such a case, if there exist activating subsets of tokens, this transition starts its production phase again, by immediately taking new tokens.

- *Transition states.* Function h describes the t-state of each transition t_i by the ordered pair $h(t_i) = (u_i, w_i)$ (Definition 21).

- *Token dynamics.* Each token in a place has its own lifetime counter κ_x that starts at zero when a token is created in a place. The lifetime κ_x of each token increases with the flow of time τ. A token matures when its lifetime is equal to or exceeds the maturity value γ^L_j of place p_j. A token is removed from its place when its lifetime exceeds the maximum value γ^U_j of that place. Regarding tokens, as explained in the description of Definition 18, it is a matter of exactly which tokens will be taken from pre-set ${}^\bullet t$ at the start of the production phase. At the beginning of the production phase, transition t takes the appropriate number of tokens from ${}^\bullet t$. Three possible token-selection strategies are to select the youngest, the oldest or randomly chosen tokens. For the proposed net, it is assumed that the oldest tokens are consumed when the production phase starts for any transition.

When the p-state of the net is described at a given time (by multiset M), and if there exists transition t that is active and can start producing tokens, one can say that

- transition t will take a number of tokens from the places, resulting in the transformation of multiset M into M' ($M \rightarrow M'$), where $M' = M - M^{t\ominus}$ (Definition 15);

- then, after some time, t will produce new tokens (represented in multiset $M^{t\oplus}$), transforming multiset M' into M'', ($M' \rightarrow M''$) where $M'' = M' + M^{t\oplus}$ (Definition 14).

Some time must pass between the state described by multiset M' and that described by M'' (this is the so-called token production time). However, from the moment the production begins, there is no possibility to interrupt this process. A transition that is already in the production phase will always create new tokens after its production time.

Unlike in the case of other popular time nets, changing the state of other transitions that are in conflict

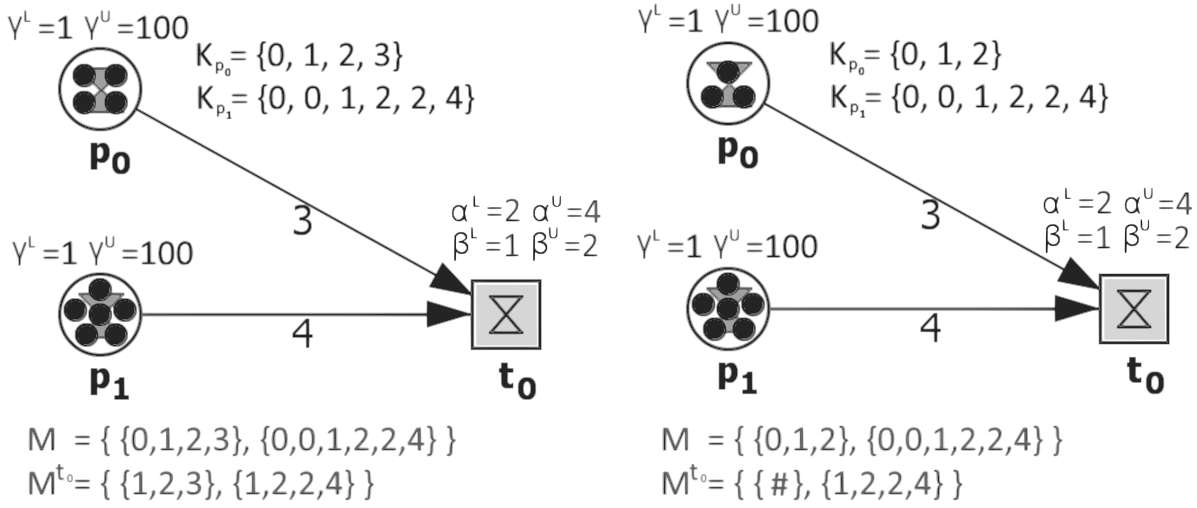


Fig. 2. Example of conditions when a transition is active or inactive.

(i.e., share a common pre-place) does not reset the activation timer (represented by u_t). For example, if t_1 and t_2 share the same place p_1, when t_1 starts or ends its production phase and there is still an activating subset of tokens for t_2 in p_1, the timer u_{t_2} does not reset and it still counts time.

The full xTPN state space is significantly larger than the state space of basic TPNs, because it combines token ages in places, transition activation and production timers that determine discrete transition phases. Therefore, exhaustive model checking based on direct state-space exploration may be applicable only to restricted or relatively small xTPN models. For larger models, especially biological or biochemical ones, simulation-based analysis becomes an important practical tool for studying the behavior of the model.

It should be emphasized, however, that simulation-based validation is not the same as formal verification and cannot replace it. Simulation can support model validation by generating possible behaviors, comparing scenarios and estimating quantitative characteristics of places and transitions. Formal verification, on the other hand, requires exhaustive or symbolic reasoning about all relevant reachable states. In this paper we define the xTPN formalism and its operational semantics, but we do not provide methods for state space analysis.

Remark 3. (*Consistency of place and transition-based timing*) In the proposed net, time assigned to places (given by function J and values γ_p^L, γ_p^U) and transition time values (given by function I and values $\alpha_t^L, \alpha_t^U, \beta_t^L, \beta_t^U$) act on different components of state $z = (m, h)$. Place timing is represented in the p-state given by m (token ages), while transition timing is represented in the t-state given by function h, i.e., timers (u_t, w_t) for each transition.

The interaction of these two time mechanisms is explicit and limited to transition activation. Place times determine whether activating subsets of tokens exist, and thus whether a transition can be active. In contrast, the transition timing governs the evolution of the activation timer u_t and the production timer w_t according to the transition state.

In particular, once a transition enters the production phase and tokens are consumed from its pre-set ${}^\bullet t$, no further changes in the activation conditions are checked until the production phase ends. Therefore, token production cannot be interrupted by later changes in token ages in places from ${}^\bullet t$. This clarifies the separation between place-based and transition-based timing. They interact through the activation condition, but they do not assign conflicting timing roles to the same transition state.

Remark 4. (*Soundness proof obligations*) The above definitions also indicate how a proof of soundness for an xTPN could be formulated. In this context, soundness means that the operational semantics preserves well-formed states and does not generate contradictory timing interpretations. State $z = (m, h)$ is well formed if the following conditions are satisfied:

1. Every token age stored in $m(p)$ belongs to the interval $[0, \gamma_p^U]$ for its place p.
2. For every transition t, the value $h(t)$ is in one of the three admissible forms: $(\#, \#)$, $(u_t, \#)$, or $(\#, w_t)$.
3. If $h(t) = (u_t, \#)$, then $0 \leq u_t \leq \alpha_t^U$ and the transition is compatible with the activation condition from Definition 24.

4. If $h(t) = (\#, w_t)$, then $0 \leq w_t \leq \beta_t^U$ and the transition is in the production phase, where activation conditions are not re-evaluated until production ends.

Under admissible time elapse, token ages and active timers are updated without violating their corresponding time limits, i.e., tokens exceeding their upper lifetime bounds are removed according to the semantics.

Starting production removes only tokens belonging to the activating subset, changes the transition state from $(u_t, \#)$ to $(\#, 0)$, and therefore preserves the condition that at most one timer of a transition be active. Ending production adds new tokens of age zero to post-places and changes the transition state from $(\#, w_t)$ to either active or inactive, depending on the current activation condition. Thus, each elementary semantic operation is designed to transform a well-formed state into another well-formed one.

A complete formal proof of this preservation property, together with that of soundness and consistency for the full xTPN class with α, β, and γ intervals present simultaneously, is not included in this paper. We treat it as an open research problem and an important direction for future work.

3. Biological example

3.1. Biological background. Gene expression in eukaryotic cells is a multi-stage process that transforms information stored in genes into functional proteins. The process called transcription begins when a promoter region is accessible, allowing initiation and elongation of an RNA transcript. The primary transcript (pre-mRNA) undergoes maturation while a parallel quality-control pathway may recognize defective transcripts and target them for degradation. Properly processed mRNA is then transferred from the nucleus to the cytoplasm, where it serves as a template for translation, producing a polypeptide chain that must fold and often undergo additional maturation before becoming a functional protein (Lodish *et al*., 2021).

Many genes encoding regulatory proteins are controlled by negative feedback to stabilize expression and prevent excessive production. Once the newly synthesized protein changes into an active form, it may inhibit further transcription of its own gene, for example, by binding to regulatory sites near the promoter, thus reducing the probability of further transcription initiation. This loop naturally produces alternating phases of production and repression: transcription continues when the promoter region is not occupied/suppressed. However, when the active regulator accumulates, promoter binding becomes more likely and the transcription is halted. As a result, protein levels are reduced, which eventually allows the transcription to resume (Kageyama *et al*., 2007).

The example considered in this section is based on this general scenario: it represents gene expression with negative feedback. The resulting structure is shown in Fig. 3.

3.2. Modeling motivation and the net structure. The purpose of the example is to demonstrate, on a biologically meaningful system, where the combined timing elements of an xTPN are useful.

The model has been created in Holmes (Radom *et al*., 2017), a tool for creation, simulation and analysis of different types of Petri nets, including classical, timed, functional or hybrid variants. As of the 2024 version, Holmes also provides dedicated support for xTPN-based models, allowing users to create such nets from scratch and to simulate their behaviors. In particular, the simulator can estimate average numbers of tokens in places, average times of transition inactivity, activity and production, as well as approximate upper bounds for the number of tokens in timed places. It supports the analysis of transition activity/production intervals. An optional mass-action kinetics simulation is also available, making the simulation more sensitive to the availability of tokens in a pre-set of a given transition. The figures presented in this section are based on simulation results obtained in Holmes.

From a modeling perspective, the net is hybrid, i.e., some transitions use both time intervals (activation α and the production β), while others only one of them. Similarly, not all places have time limits γ. This design is intentional. In biological applications, time data are often incomplete and unevenly distributed between different elements. For some processes, one may know when activation can begin, for some others how long production lasts, and for others only that the corresponding molecular entity remains useful for a limited time. Requiring the whole model to follow a single timing convention would therefore be unnecessarily restrictive. The example demonstrates that an xTPN can combine these different forms of time knowledge within a single coherent formalism.

In the model considered, transitions t_2, t_3, t_4 and t_5 use both timing intervals and exhibit the full behavior of xTPN transitions. Most of the remaining ones employ only one time range and may therefore be viewed as special cases corresponding to basic timed-Petri net classes. In particular, transitions t_0 and t_8 use only α intervals, which is a deliberate design choice that enables the representation of competition and delayed feedback in a biologically meaningful way. A similar distinction applies to places. Formally, places intended to behave classically can be interpreted as having $\gamma^L = 0$ and $\gamma^U = \infty$, while timed places preserve the full xTPN semantics.

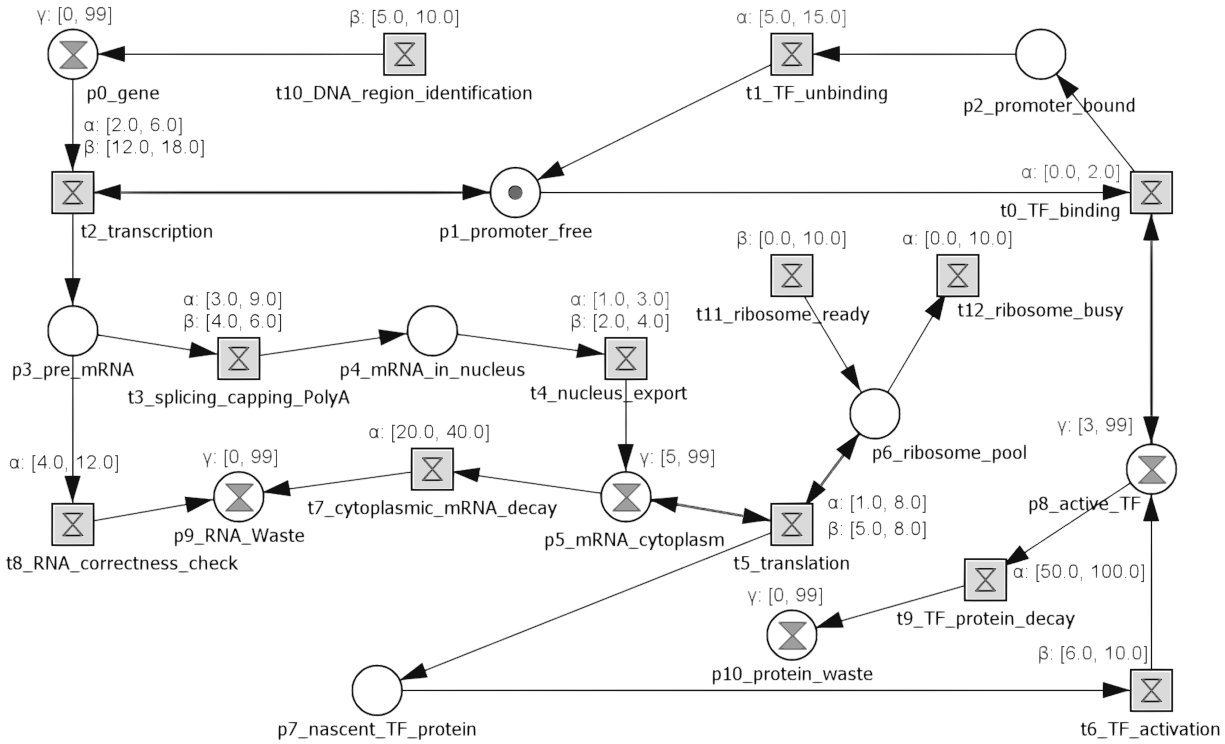


Fig. 3. Example model of gene expression with negative transcription feedback. Places and transitions are annotated with their timing parameters. Circles without hourglasses represent places without time constraints.

3.3. Conflict between pre-mRNA processing and the correcting mechanism. One of the important parts of the model represents a conflict between further processing of pre-mRNA (path $p_3 \to t_3$) and the rejection of defective transcripts (path $p_3 \to t_8$). A token appearing in p_3 can be consumed by transition t_3, representing further processing of pre-mRNA, or by transition t_8, representing RNA "quality control" in a nucleus (and then degradation). This fragment illustrates the benefit of separating the activation interval α from the production interval β.

The probability that a pre-mRNA token follows the productive or the degradation pathway depends on which of the two competing transitions first ends being active. In the present model, this conflict is caused by intervals α of t_3 and t_8, while the duration of the production process is given by the β interval of t_3. This reflects a distinction between the time required before RNA processing can start and that necessary for it to finish. As a result, assigned α intervals govern the ratio between molecules that are degraded and those that later become functional proteins.

In order for this behavior to be reproduced, additional transitions would be required if only TPN or DPN nets were available. If t_3 were represented only as a TPN transition, then the entire duration of pre-mRNA processing would have to be encoded as interval α. In such a case, t_8 would often gain an unrealistically strong competitive advantage, because a token in p_3 would remain available for surveillance throughout the waiting period. In contrast, if t_3 were represented only as an ITPN-like transition, with the entire duration of the process encoded in β, then the token in p_3 would be taken immediately, effectively preventing t_8 from competing at all. The feasible solution would require an additional place and extra TPN-like transition to separate activation and production in t_3. The xTPN avoids this problem by combining these two behaviors within time ranges of a single transition.

3.4. Negative feedback and promoter repression. A second important feature of the model appears in the negative-feedback loop responsible for temporary inhibition of the transcription process (transition t_2). Once the first mature regulatory protein appears in p_8, transition t_0 may remove the promoter token from p_1, leading to cycle $t_0 \to p_2 \to t_1 \to p_1$. Because transition t_2 requires the promoter represented by p_1, the presence of an active protein in p_8 effectively suppresses the production of new transcripts. As long as a token remains in p_8, this cycle prevents the continued expression of the gene.

The simulation results shown in Fig. 4(a) illustrate this behavior. When tokens are present in p_8, the *average* number of tokens in p_1 remains close to zero, because the promoter token circulates in the repression

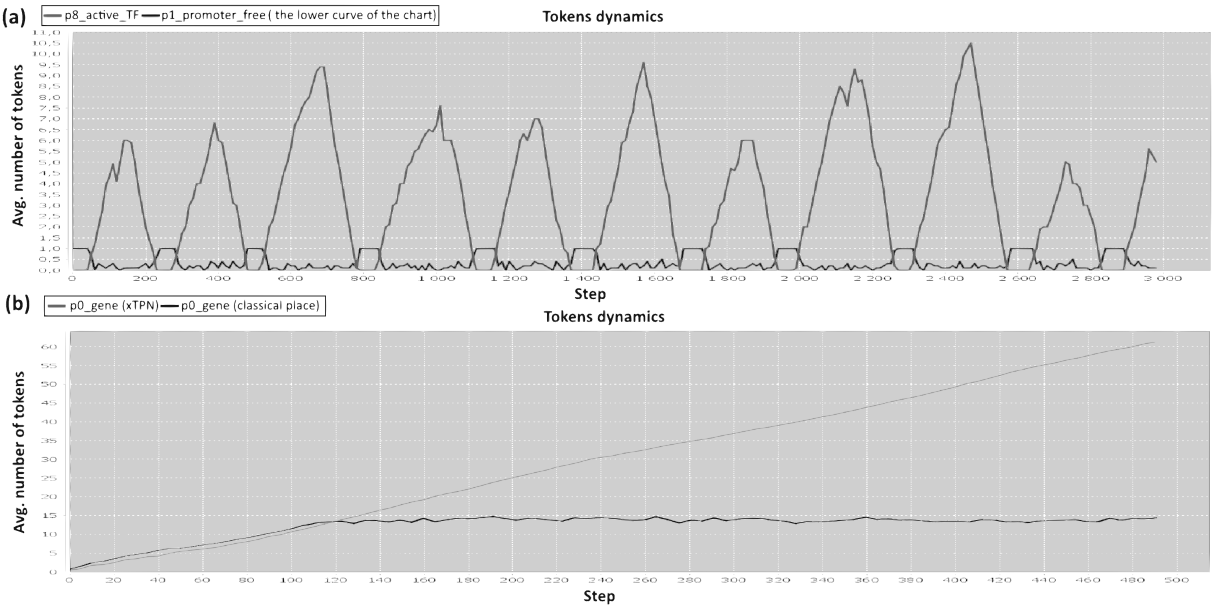


Fig. 4. Simulation-based illustration of selected model behaviors: average token counts in p_8 and p_1, showing alternating phases of transcriptional repression and promoter availability (a), bounded token count in timed place p_0 and nearly linear tokens accumulation in the classical p_0 without lifetime constraints (b).

loop and is therefore not available for transition t_2. As a side note, it should be stated that, in fact, the number of tokens in p_1 is almost always zero during transcription inhibition, whereas the chart shows averaged results of the simulation repeated many times, resulting in non-integer values in Fig. 4. Once the tokens in p_8 disappear, one stays in p_1, enabling transcription again. This alternating behavior repeats in successive intervals and produces a pattern which is consistent with a self-inhibitory regulatory system.

3.5. Role of timed places. The feedback mechanism described above is closely related to another essential feature of the xTPN, namely, timed places. In the model, place p_8 represents active protein molecules whose biological effect persists for a finite time even when their removal by transition t_9 is slow. In practice, the disappearance of these tokens is determined not only by the consuming transition but also mainly by the finite lifetime of tokens in p_8. This allows the model to express directly that regulatory proteins may remain functional for some time and then vanish as a consequence of age-dependent degradation. Capturing the same effect by transitions alone would require additional modeling constructs and would obscure the intended interpretation.

A clear illustration of the role of timed places is provided by place p_0, as shown in Fig. 4(b). In the xTPN model, p_0 is a timed (xTPN) place, so its token count remains bounded. By contrast, when the timing constraints of p_0 are removed, so that it is treated as an untimed place, the number of tokens grows almost linearly over time. This comparison directly shows that timed places allow the model to represent the disappearance or degradation of entities as a function of time without introducing additional auxiliary transitions. Reproducing the same behavior with transitions from component nets alone (TPN or ITPN) would substantially increase the size and complexity of the model. Emulation of time-places would be even more difficult or in some cases even impossible to obtain as a full xTPN behavior.

Timed places also make it possible to represent the delayed usefulness (maturity) of tokens (γ^L) after their initial creation. In the example, this behavior is captured in places such as p_5 and p_8, for which $\gamma^L > 0$. A newly produced token is therefore not immediately available for consumption or activation. This delay belongs to the state of a token after its production and is therefore represented more naturally by timed places than by transition-based timing alone.

3.6. Interpretation and scope of the example. The time parameters of the model were chosen illustratively, that is, not as a result of a quantitative fitting to experimental measurements but in such a way that their relative ordering reproduces plausible qualitative behavior of transcription, RNA processing and negative feedback through promoter repression. The goal of the example is to demonstrate that an xTPN can represent, within a single compact model, biologically meaningful combinations of activation and production delays, token maturation and

Table 1. Possible transformations of extended time Petri net elements into simpler ones. It should be noted that transformation for a place-transition Petri net is denoted with a star. This is because, if $\alpha^L = \alpha^U = 0$, then such a transition must technically fire immediately, which is not enforced in most classical Petri nets.

Net element	Time parameters settings	Result
Place	$\gamma^L = 0, \gamma^U = \infty$	Classical place
Transition	$\alpha^L, \alpha^U \in \mathbb{Q}_0^+, \beta^L = \beta^U = 0$	TPN transition
Transition	$\alpha^L = \alpha^U = 0, \beta^L, \beta^U \in \mathbb{Q}^+$	ITPN transition
Transition	$\alpha^L = \alpha^U = 0, \beta^L, \beta^U \in \mathbb{Q}^+, \beta^L = \beta^U$	DPN transition
Transition	$\alpha^L = \alpha^U = 0, \beta^L = \beta^U = 0$	Classical transition*

its finite lifetime. For this reason, the proposed case study addresses two issues. First, it shows that the model can be implemented and simulated in dedicated software. Second, it demonstrates that the coexistence of intervals α, β and γ not only is a formal generalization but also has direct modeling consequences that would be difficult to reproduce only by using TPN-like or ITPN-like nets.

4. xTPN transformation

In this section we describe selected transformation properties of an xTPN with an accompanying proof.

One of useful features of the described Petri nets is the possibility of transforming their elements to the ones that are known in other types of time Petri nets. This makes it possible to create a model of a system in which different types of transitions and places with time parameters can exist simultaneously. Even more important is the fact that different types of time data that describe such a system can be used in a single model without the need for additional adjustments. In Table 1 possible transformations are given, which depend on the proper assignment of values to specific time variables α, β and γ.

To transform the xTPN time place into a classical one (that is, without time constraints for tokens), $\gamma_p^L = 0$ and $\gamma_p^U = \infty$. Tokens in such a place will never expire and can be used to activate a transition immediately when they are created. Formally we can write that a *reduced* extended time Petri net $\mathcal{Z}_{red}$ is given, where, for every transition $t \in T$, $\beta_t^L = \beta_t^U = 0$, and for all places $p \in P$, the token age constraints are irrelevant, that is, $J(p) = [0, \infty)$. Furthermore, in $\mathcal{Z}_{red}$ there are no inhibitor nor read arcs.

Theorem 1. (Reduction of an xTPN for $\beta \equiv 0$) *Let $\mathcal{T} = (\mathcal{N}, I)$ be a time Petri net. $\mathcal{Z}_{red}$ and $\mathcal{T}$ are equivalent with respect to their state spaces, that is, every p- and t-state of $\mathcal{Z}_{red}$ corresponds to a respective state of $\mathcal{T}$. The sets of reachable states in both nets are the same, and in each state the sets of enabled transitions are identical.*

Proof. Let a state of $\mathcal{Z}_{red}$ be denoted by $z = (m, h_x)$, where $m : P \to M$ is a p-state and $h_x : T \to (\mathbb{R}_0^+ \cup \{\#\}) \times (\mathbb{R}_0^+ \cup \{\#\})$ is a t-state, $h_x(t) = (u_t, w_t)$. We assume that (i) for every transition t: $\beta_t^L = \beta_t^U = 0$, (ii) for every place p: $J(p) = [0, \infty)$ and (iii) $F^{inh} \cup F^{read} = \emptyset$.

Mapping states and transition activity. We define the projection of a p-state into token counts in each place as $\widehat{m}(p) := |m(p)|$ for $p \in P$, that is, the cardinality of multiset K_p determined by $m(p)$. Let us also define the mapping of the t-state of $\mathcal{Z}_{red}$ into $\mathcal{T}$ notation by

$$\Phi(h_x)(t) := \begin{cases} u_t & \text{if } h_x(t) = (u_t, \#) \\ & (t \text{ enabled and not producing}), \\ \# & \text{if } h_x(t) = (\#, \#) \\ & (t \text{ is not enabled}). \end{cases}$$

Since $\beta_t \equiv 0$, a state $h_x(t) = (u_t, w_t)$ with $w_t \in \mathbb{R}^+$ does not exist that is, the start and end of production is instantaneous (that time is always zero: $w_t = 0$).

Consequently, Φ is defined for all reachable states. With $J(p) = [0, \infty)$ and no read or inhibitor arcs, the enabling condition for t in $\mathcal{Z}_{red}$ reduces to $\forall p \in {}^\bullet t : \widehat{m}(p) \geq V(p, t)$, i.e., exactly the same condition as in $\mathcal{T}$ for the p-state $\widehat{m}$. Therefore, t is enabled in $(m, h_x) \iff$ t is enabled in $(\widehat{m}, \Phi(h_x))$.

Consistency of time elapse. A time elapse by $\tau \geq 0$ in an xTPN has two effects. First, it increases every element of multiset M defined by $m(p)$ by τ. However, since $J(p) = [0, \infty)$, the time elapse will never remove tokens, so in that case $\widehat{m}$ does not change. Since maturity time of tokens is also assumed to be zero, the time elapse effect on places has no impact on the number of tokens and activation conditions. Secondly, time elapse increases u_t by τ for every enabled transition (those with $h_x(t) = (u_t, \#)$), leaving the other values unchanged.

In a TPN, time elapse acts identically on the p-state (that is, there is no change in number of tokens) and on the t-state (clocks of enabled transitions are increased). In addition, the upper bound on the allowed enabling time applies: $u_t \leq \alpha_t^U$. Its violation is disallowed both in $\mathcal{Z}_{red}$ and in $\mathcal{T}$ (with $I_2(t) = \alpha_t^U$).

Discrete step in both nets. In $\mathcal{Z}$ with $\beta_t \equiv 0$, starting the production of t at time $u_t \in [\alpha_t^L, \alpha_t^U]$ has two effects. It removes from every place $p \in {}^\bullet t$ exactly $V(p,t)$ tokens. Under the assumption that $J = [0, \infty)$, the choice of taking the oldest tokens reduces to decreasing the multiplicity in the place, i.e., $\widehat{m}(p) \mapsto \widehat{m}(p) - V(p,t)$. The second effect is that, immediately after token removal, the transition adds to every $q \in t^\bullet$ exactly $V(t,q)$ tokens: $\widehat{m}(q) \mapsto \widehat{m}(q) + V(t,q)$. Therefore, the p-state for every $p \in P$ changes as follows:

$$\widehat{m}'(p) = \begin{cases} \widehat{m}(p) - V(p,t) & \text{if } (p,t) \in F \\ \quad \text{and } (t,p) \notin F, \\ \widehat{m}(p) + V(t,p) & \text{if } (t,p) \in F \\ \quad \text{and } (p,t) \notin F, \\ \widehat{m}(p) - V(p,t) + V(t,p) \\ \quad \text{if } (p,t) \in F \text{ and } (t,p) \in F, \\ \widehat{m}(p) \text{ otherwise.} \end{cases}$$

This is exactly the same as a single firing in $\mathcal{T}$. As for the t-state, transition t finishes being enabled; however, because of $w_t = 0$, the production is immediate. Then it becomes either not enabled (then $\Phi(t)$ yields #) or, if t remains enabled, it receives a new clock $\Phi(t) = u_t = 0$ (in net $\mathcal{T}$: $h(t) = 0$).

Bisimulation of states. Let us define the state mapping from an xTPN to a TPN by function $\Psi(m, h_x) := (\widehat{m}, \Phi(h_x))$. From the previous paragraphs it follows that

- if the xTPN state changes due to time elapse or a transition firing, denoted as $(m, h_x) \xrightarrow{\tau \text{ or } t} (m', h'_x)$, then for the TPN the analogous state change is $\Psi(m, h_x) \xrightarrow{\tau \text{ or } t} \Psi(m', h'_x)$;
- conversely, any time elapse or transition firing in the TPN has an exact counterpart in the xTPN, with the same effect on $\widehat{m}$ and with same clock updates.

Since every step in one net can be simulated by a step in the other (and vice versa), the two nets have the same reachable states, and in paired states the same transitions are enabled. ■

For an ITPN/DPN, a similar transformation can be proved analogously by exchanging the role of α and β. Transformation of an xTPN into a P-TPN is possible, but with an additional condition. When both values α and β are set to zero, the xTPN transition activates, fires and produces tokens immediately. This is not the behavior of a classical PN transition, which, being enabled, is not forced to fire. If such a condition changes, the default behavior of xTPN transitions is assumed, and transformation into a P-TPN is also possible.

5. Conclusions

The proposed extended time Petri net has many more interesting properties and a more detailed description of some of them is beyond the scope of this paper. Two features in particular, in our opinion, suggest that the proposal of a new time net combining the features of already known ones can be a good and interesting idea. First of all, it is often the case that, when trying to create a model of a certain system, the net that is selected must best fit the available data for that system. However, it is often necessary to match the data and the net elements with each other because the specific net perhaps fits well, but not ideally, to the type of data available. For example, transitions in a TPN remain active for a certain period of time before firing; when they fire, tokens are taken from the input places while new ones are simultaneously created at the output places. In a DPN/ITPN, an active transition starts the production event immediately, that is, it takes tokens, but the production of new ones requires some time. The difference is that, in a TPN, in case of a conflict between many transitions sharing common places, tokens from a given place can simultaneously participate in the activation of many transitions until they are taken. In a DPN/ITPN they cannot do so since they are consumed immediately. If the system studied has elements that require both behaviors, some conversions or simplifications must take place for that model. This, in turn, can further obscure the overall picture of the model. In the proposed extended time Petri net, a transition implicitly has both features, i.e., a time until activation and a separate production time counted when tokens have already been taken. Transformation of net elements is possible and can be done by modifying only the time data describing the transitions. Thus, there is no need to add some artificial elements to the net while transforming it to reflect either behavior. The second important feature of the new net is that transitions and places are described by two and one time intervals, respectively, rather than by a specific single value of, for example, production time, as in a DPN. One can imagine many systems in which the time of execution of certain activities cannot be reduced to one specific and deterministic value, but it can at most be given as an approximation. Biological systems in particular are often described in this way because it is almost impossible to give the precise time of occurrence of, for example, a single chemical reaction. Thus, the proposed net turns out to be a very flexible tool that can be easily adapted to the data and knowledge of some system.

The proposed xTPN formalism introduces a richer state description than simpler time Petri-net classes. As a consequence, the corresponding state space is substantially larger than in classical TPNs, and exhaustive state-space exploration or direct model checking may be computationally feasible only for restricted or relatively

small xTPN models. Therefore, the formal definitions and semantics introduced in this paper should be understood as a necessary formal basis for future development of verification methods and state-space verification. In particular, simulation-based analysis of xTPN models, including simulations performed in Holmes, can be useful for validation or a quantitative analysis, but it is not equivalent to formal verification and cannot replace exhaustive or symbolic reasoning about all relevant reachable states.

The proposed xTPN formalism should currently be viewed primarily as a modeling and simulation framework for systems in which several types of timing constraints must be represented simultaneously. Its formal syntax and semantics provide a foundation for future verification methods, but the full state space is substantially larger than in classical TPNs because it includes token ages in places, transition activation timers, transition production timers and discrete transition phases. Therefore, exhaustive state-space verification and complete model checking may be limited to relatively small or restricted xTPN models.

For larger biological or biochemical models, simulation-based validation such as repeated simulations in the program Holmes can be a practical way to study possible behaviors and estimate characteristics of the model. However, such validation is not a formal verification and cannot replace exhaustive or symbolic reasoning regarding all relevant reachable states. A comprehensive formal proof of soundness and consistency for the full xTPN model, incorporating time intervals α, β, and γ simultaneously, together with dedicated symbolic state-space abstractions and model-checking algorithms, remains an open research problem and one of the main directions for future work.

Acknowledgment

This work was partially supported by statutory funds of the Poznan University of Technology.

References

Berthomieu, B. and Diaz, M. (1991). Modeling and verification of time dependent systems using time Petri nets, *IEEE Transactions on Software Engineering* **17**(3): 259–273.

Berthomieu, B. and Menasche, M. (1983). An enumerative approach for analyzing time Petri nets, *in* R.E.A. Mason (Ed), *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France*, Elsevier Science Publishers, Amsterdam, pp. 41–46.

Bolognesi, T., Lucidi, F. and Trigila, S. (1990). From timed Petri nets to timed LOTOS, *Proceedings of the IFIP WG6.1 10th International Symposium on Protocol Specification, Testing and Verification, Ottawa, Canada*, pp. 395–408.

Boucheneb, H., Barkaoui, K., Xing, Q., Wang, K., Liu, G. and Li, Z. (2022). Time based deadlock prevention for Petri nets, *Automatica* **137**: 110119.

Bourdil, P.-A., Berthomieu, B., Zilio, S.D. and Vernadat, F. (2016). Symmetry reduction for time Petri net state classes, *Science of Computer Programming* **132**: 209–225.

Coolahan, J.E. and Roussopoulos, N. (1983). Timing requirements for time-driven systems using augmented Petri nets, *IEEE Transactions on Software Engineering* **9**(5): 603–616.

Dubois, T., Larsen, K.G. and Srba, J. (2025). Statistical model checking of stochastic timed-arc Petri nets, *in* E. Amparore and Ł. Mikulski (Eds), *Application and Theory of Petri Nets and Concurrency*, Springer Nature Switzerland, Cham, pp. 174–196.

Jacobsen, L., Jacobsen, M., Møller, M.H. and Srba, J. (2011). Verification of timed-arc Petri nets, *in* I. Černá *et al.* (Eds), *SOFSEM 2011: Theory and Practice of Computer Science*, Springer, Berlin/Heidelberg, pp. 46–72.

Jiang, W., Zhou, K.-Q., Sarkheyli-Hägele, A. and Zain, A.M. (2022). Modeling, reasoning, and application of fuzzy Petri net model: A survey, *Artificial Intelligence Review* **55**(8): 6567–6605.

Kageyama, R., Ohtsuka, T. and Kobayashi, T. (2007). The Hes gene family: Repressors and oscillators that orchestrate embryogenesis, *Development* **134**(7): 1243–1251.

Khansa, W., Denat, J.-P. and Collart-Dutilleul, S. (1996). P-time Petri nets for manufacturing systems, *International Workshop on Discrete Event Systems, WODES, Edinburgh, UK*, pp. 94–102.

Leclercq, L., Lime, D. and Roux, O.H. (2023). A state class based controller synthesis approach for time Petri nets, *in* L. Gomes and R. Lorenz (Eds), *Application and Theory of Petri Nets and Concurrency*, Springer Nature Switzerland, Cham, pp. 393–414.

Lodish, H.F., Berk, A., Kaiser, C., Krieger, M., Bretscher, A., Ploegh, H.L., Martin, K.C., Yaffe, M.B. and Amon, A. (2021). *Molecular Cell Biology*, 9th Edn, Macmillan International Higher Education, New York.

Merlin, P.M. (1974). *A Study of the Recoverability of Computing Systems*, PhD thesis, University of California, Irvine, Available online at `https://escholarship.org/content/qt1p80c4fg/qt1p80c4fg.pdf`.

Mizuta, A., Ge, Q.-W. and Matsuno, H. (2017). Dependent shrink of transitions for calculating firing frequencies in signaling pathway Petri net model, *Algorithms* **10**(1), Article no. 4.

Moigne, M. L., Mahulea, C., Faraut, G., Bernardi, S., Albareda, J. and Castán, L. (2025). Probabilistic timed Petri nets for clinical pathway design and analysis: A case study, *Discrete Event Dynamic Systems* **35**(3): 205–231.

Murata, T. (1989). Petri nets: Properties, analysis and applications, *Proceedings of IEEE* **77**(4): 541–580.

Petri, C.A. (1962). *Communication with Automata*, PhD thesis, Schriften des Instituts fur Instrumentelle Mathematik,

Bonn, Available online at https://scispace.com/pdf/communication-with-automata-2v17tps1p5.pdf.

Popova-Zeugmann, L. (1993). Petri nets with time restrictions, *Systems Analysis Modelling Simulation* **13**(1): 13–20.

Popova-Zeugmann, L. (2013). *Time and Petri Nets*, Springer, Berlin.

Popova-Zeugmann, L. and Pelz, E. (2021). Algebraical characterisation of interval-timed Petri nets with discrete delays, *Fundamenta Informaticae* **120**(3–4): 341–357.

Radom, M. and Formanowicz, P. (2022). Algorithmic aspects of extended time Petri nets, *in* A. Świerniak and J. Krystek (Eds), *Automation of Discrete Processes*, Vol. 2, Silesian University of Technology Press, Gliwice, pp. 91–102, (in Polish).

Radom, M., Rybarczyk, A., Szawulak, B., Andrzejewski, H., Chabelski, P., Kozak, A. and Formanowicz, P. (2017). Holmes: A graphical tool for development, simulation and analysis of Petri net based models of complex biological systems, *Bioinformatics* **33**(23): 3822–3823.

Ramchandani, C. (1974). *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets,* PhD thesis, Massachusetts Institute of Technology, Cambridge, Available online at https://dspace.mit.edu/entities/publication/6a954095-66b6-457e-bc47-0fdd50933aa1.

Sifakis, J. (1980). Performance evaluation of systems using nets, *in* W. Brauer (Ed.), *Net Theory and Applications*, Springer, Berlin/Heidelberg, pp. 307–319.

Sobrino, A., Garrido-Merchán, E.C. and Puente, C. (2021). Fuzzy stochastic timed Petri nets for causal properties representation, *New Mathematics and Natural Computation* **17**(3): 633–653.

Szawulak, B., Radom, M. and Formanowicz, P. (2022). Comparing Petri net-based models of biological systems using Holmes, *Bioinformatics* **38**(19): 4652–4653.

van der Aalst, Wil M. P.and Leemans, S.J.J. (2025). Learning generalized stochastic Petri nets from event data, *in* N. Jansen *et al.* (Eds), *Principles of Verification: Cycling the Probabilistic Landscape*, Springer Nature Switzerland, Cham, pp. 3–17.

Wegener, J.-T. and Popova-Zeugmann, L. (2009). Petri nets with time windows: A comparison to classical Petri nets, *Fundamenta Informaticae* **93**(1–3): 337–352.

Yoshimasa, M., Chen, L., Qi-Wei, G., Matsuno, H. and Miyano, S. (2010). On determining firing delay time of transitions for Petri net based signaling pathways by introducing stochastic decision rules, *In Silico Biology* **10**: 49–66.

Zhou, J., Wang, J. and Wang, J. (2019). A simulation engine for stochastic timed Petri nets and application to emergency healthcare systems, *IEEE/CAA Journal of Automatica Sinica* **6**(4): 969–980.

Zilio, S.D., Fronc, L., Berthomieu, B. and Vernadat, F. (2014). Time Petri nets with dynamic firing dates: Semantics and applications, *12th International Conference, FORMATS, Florence, Italy,* pp. 85–99.



Marcin Radom received his PhD in computer science from the Poznan University of Technology in 2011 and has been working as an assistant professor in the Faculty of Computing Science there since 2012. He also works in the Institute of Bioorganic Chemistry, Polish Academy of Sciences. His research focuses on DNA sequencing, precisely the complexity problems connected with various sequencing methods. Other areas of his interest include systems biology (i.e., modeling complex biological systems using Petri nets and model analysis issues) and programming in Java and C#.

Piotr Formanowicz received his PhD degree in 2000 and his habilitation qualification in 2005 from the Poznan University of Technology. In 2015, he was granted the professorial title. He is a full professor at the Institute of Computing Science, Poznan University of Technology. His research interests include combinatorial optimization, scheduling theory, computational complexity, graph theory, computational biology and systems biology.

Received: 6 December 2025
Revised: 2 April 2026
Re-revised: 5 June 2026
Accepted: 9 June 2026