BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 66, No. 1, 2018
DOI: 10.24425/119060

Some aspects of modeling and analysis of complex biological systems using time Petri nets

J. OLSZAK[1, 2], M. RADOM[1, 3], and P. FORMANOWICZ[1, 3]*

[1] Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland
[2] Lhasa Limited, Granary Wharf House, 2 Canal Wharf, Holbeck, Leeds LS11 5PY, U.K.
[3] Institute of Bioorganic Chemistry, Polish Academy of Sciences, ul. Z. Noskowskiego 12/14, 61-704 Poznan, Poland

Abstract. Models of complex biological systems can be built using different types of Petri nets. Qualitative nets, for example, can be successfully used to obtain a model of such a system and on its basis a structure-based analysis can be performed. Time is an important factor influencing a whole biological system behaviour and in many cases it should be considered during building a model of such a system. In this paper various types of time Petri nets have been described and methods for studying corresponding models have been discussed. In particular, an algorithm using time parameters to enhance t-invariants based analysis is proposed. This algorithm allows for calculation of the minimal and maximal numbers of tokens (respectively, for an optimistic and pessimistic case) in particular places necessary to assure that all transitions from a given t-invariant support will be able to fire. Additionally, to address the problem of the proper assignment of time values to transitions, the known methods for calculation and evaluation of such time parameters based on the net structure have also been discussed.

Key words: biological systems, time Petri nets, t-invariants.

1. Introduction

Rapid development of biological sciences, which can be observed since 1990s, and an enormous increase of an amount of biological data of various types caused a growing belief that in order to fully understand the nature and functionality of living organisms, a new approach in research is needed. According to the standard approach, some basic building blocks of the organisms are analyzed in great detail but much less attention is paid to interactions among them. Yet during the last two decades it has seemed more and more evident that these interactions play crucial roles. Moreover, the basic building blocks are connected by a very dense and complex network of such interactions and the structure of this net determines many fundamental properties of an organism (or its functional blocks like organs, tissues, cells etc.). Despite that the standard approach resulted in many spectacular discoveries in all areas of biological sciences, it seems to have serious limitations and it is not well suited to the analysis of the already mentioned interaction networks.

Since living organisms and their functional blocks are complex systems of interacting basic elements they should be analyzed using methods appropriate for such systems [1, 2]. In general, such methods are developed in the area of system sciences. However, many of those methods have been constructed in the context of technical systems. Often, they can be applied to the analysis of biological systems but they have a certain specificity which causes that direct applications of methods developed to the analysis of systems of other types may be insufficient. In other words, often it is necessary to adapt such methods to the specificity of biological systems or even to develop new methods. This is an area of a relatively new branch of sciences called systems biology, which aims at analyzing biological phenomena as complex systems [3].

Such an analysis is based on a formal model of the system. The model can be expressed in a language of some branch of mathematics; usually, differential equations are used for this purpose. Despite their great expressive power, they also have limitations in the context of modeling of biological systems. They follow from the necessity of determining exact values of some parameters which correspond to some quantitative properties of the modeled system. In practice, determining these values is usually very difficult or even impossible. Hence, some other methods of building models of biological systems are looked for. The ones based on graph theory or mathematical objects similar to graphs seem to be especially promising. One of the reasons for this is the fact that graphs are well suited to describe a structure of dependencies among the building blocks of the biological system and their intuitive graphical representation is very helpful in understanding the structure of these dependencies. Moreover, there are various methods for formal analysis of graph-based models.

Despite that Petri nets are not graphs, they have a structure of a directed bipartite graph [4]. Hence, models of biological systems expressed in the language of Petri net theory have many advantages of graph-based models and, in addition, they

*e-mail: piotr@cs.put.poznan.pl

Manuscript submitted 2017-01-31, revised 2017-09-01, initially accepted for publication 2017-10-24, published in February 2018.

allow to model and analyze a dynamics of these systems. It is possible thanks to flows of tokens which correspond to flows of substances, information etc. through the modeled system. Hence, these nets are recently considered as a very promising mathematical tool for systems biology [5, 6]. One of the examples can be the analysis of metabolic pathways by using Petri net (cf. [7–10]).

Models based on classical Petri nets are qualitative, which means that primarily they describe a structure of the modeled system. It could be seen as a serious limitation but in the case of biological systems their structure is usually crucial for their functionality. It means that based on qualitative models it is often possible to discover some important properties of a system of this type. Such properties usually follow from interactions between subprocesses in the modeled system. These interactions can be found by analyzing of the structure of the net (the subprocesses correspond to t-invariants). Moreover, there is a number of extensions of classical Petri nets which allow to include in models based on them qualitative information of various types. Using such extensions, it is possible to increase precision of models and as a consequence, it allows a more precise analysis of the modeled system [11].

Among these extensions there are time Petri nets (in fact, several types of them) [12–14]. Such nets allow for describing durations of processes occurring in the modeled system as well as time dependencies between them. It is a very important feature since time is a crucial component of every physical system and it can considerably influence its behavior. Hence, time Petri nets greatly increase the expressive power of models based on them in comparison to those expressed in the language of classical Petri net theory. As a consequence, they significantly increase the usefulness of Petri nets in the area of modeling and analysis of biological systems. Literature references where Petri nets with time have been used include [15–18].

In this paper there will be briefly described various types of time Petri nets and selected methods of analysis of biological systems' models based on them. The structure of the paper is as follows. In Section 2, basic notions of classical Petri net theory are presented and definitions of time Petri nets are formulated. In Section 3, methods of analysis of time Petri nets based on t-invariants are discussed. In Section 4 an algorithm for calculation of a number of tokens necessary for transition firing is proposed. In Section 5 methods of estimation and modification of time parameters are presented. The paper ends with conclusions given in Section 6.

2. Classical and time Petri nets

Before definitions for Petri nets with time will be given and explained, we will formally define a classical Petri net (PN) as a base for further extensions. Definitions concerning analysis of such a net will be kept at minimum, as they can be found, e.g., in [4].

2.1. Classical Petri net. Classical Petri net can be described by Definition 1.

Definition 1. Petri net [19].
Petri net is set $N = \{P, T, f, m_0\}$, where:
P and T are finite, non-empty and disjoint sets, respectively of places and transitions,
$f : \big((P \times T) \cup (T \times P)\big) \to \mathbb{N}$ defines a set of arcs with weights being non negative integer values,
$m_0 : P \to \mathbb{N}$ is an initial marking for a net.

The structure of a Petri net is a bipartite directed graph, with vertices divided into two disjoint sets P and T, respectively called places and transitions, connected together by arcs. In places reside objects called tokens. In biological models places often represent passive components of the system, e.g., substrates or products of chemical reactions. Transitions represent some elementary subprocesses, e.g., chemical reactions. The number of tokens represent the amount of a passive component in the system a given place represents. A marking of a net is a vector which describes an exact number of tokens in each place at a given moment. It corresponds to a state of the modeled system.

When talking about a Petri net structure it is convenient to define post- and pre-places and transitions. A place p_i with an arc directed into transition t_j is a pre-place of t_j, while on the other hand, t_j can be considered as a post-transition of p_i. On the opposite, transition t_j with an arc directed into place p_i is a pre-transition of p_i, while p_i can be considered as a post-place of transition t_j. Sets of pre-places and post-places of transition t_j will be denoted as ${}^\bullet t_j$ and $t_j{}^\bullet$ respectively, while sets of pre-transitions and post-transitions of place p_i as ${}^\bullet p_i$ and $p_i{}^\bullet$.

The number of tokens can change due to a firing of an enabled transition. A transition t_j is enabled, if for every place belonging to ${}^\bullet t_j$ there are at least as many tokens as the value of a weight of the arc connecting such place with t_j. In a classical Petri net an enabled transition can fire, but not necessarily has to. The firing of transition t_j consumes the tokens from its pre-places and produces them in all its post-places. The number of tokens consumed and produced for any such a place is always equal to the weight of an arc connecting a given place with t_j [4]. For example, if transition t_j have two pre-places with weights 4 and 2 respectively and one post-place with a weight equal to 3, when it fires it will take four tokens from its first pre-place, two tokens from the second one (assuming of course there are at least as many tokens present in them) and then instantly t_j will produce three tokens in its post-place.

Formally, the transition firing can be presented as a firing rule in Definition 2:

Definition 2. Firing rule [20].
Let $N = \{P, T, f, m_0\}$ be a Petri net:
- Transition is enabled in marking m, written as $m[t\rangle$, if $\forall p_i \in {}^\bullet t_j : m(p_i) \geq f\big(p_i, t_j\big)$, else disabled.
- Transition t_j, which is enabled in m, may fire.
- When t_j in m fires, new marking m' is reached, with $\forall p_i \in P : m'(p_i) = m(p_i) - f(p_i, t_j) + f\big(t_j, p_i\big)$.
- The firing happens instantaneously and does not consume any time.

The initial marking of a net is a vector which determines the starting numbers of tokens in all places before firing of any transition. Such firing will of course change the state of the net. A state space is a set of all states of the net that can be achieved due to the firings of any transition sequence, starting from the initial marking. Such a space or its fragment can be represented in a form of, e.g., a state graph, where vertices represent the markings (state vectors) and arcs represent changes of tokens distributions due to transition firings. Studying the state space can provide detailed knowledge about the behavior of the system model which the net represents. However, if the search space is too large or even infinite, its realistic analysis can only be performed using approximate methods.

A Petri net can be represented as an incidence matrix. In such a matrix $A = [a_{ij}]_{n \times m}$ for a Petri net having n places and m transitions, every entry $a_{i,j}$ contains an integer number being a difference between the numbers of tokens residing in place p_i before and after firing transition t_j. t-invariant is a vector $x \in \mathbb{N}^m$ such that $A \cdot x = 0$. A support of t-invariant x is set $s(x)$ of transitions which correspond to positive entries of x, i.e., $s(x) = \{t_j : x_j > 0; j = 1, 2, \ldots, m\}$. Firing all the transitions from $s(x)$ a proper number of times (defined by the non-zero entries of x) does non change the marking of the net. The net is covered by t-invariants if every transition belongs to a support of at least one t-invariant. In the case of a Petri net being a model of a biological system t-invariants correspond to more complex subprocesses occurring in the system which do not change its state. t-invariants are very useful for an analysis of Petri net based models of biological systems (cf. [8, 9, 21]). At the end of this section, elementary flux modes in metabolic network analysis should be mentioned [22]. Elementary flux modes represent minimal sets of enzymes which can operate at steady state. Also, no other flux modes at a steady state are proper subsets of elementary flux modes. Minimal t-invariants in Petri nets are elementary flux modes counterparts, however, the latter are more general because reversible reactions are allowed [23]. Other relevant information can be found in, e.g., [6, 24].

2.2. Petri nets with time. Extensions of Petri nets where time is taken into account can be defined in many different ways. The two most popular ones are time Petri nets (TPN) and time Petri nets (DPN). Below, in Definition 3 time Petri net is presented.

Definition 3. Time Petri net TPN [14]
Time Petri net is a set $N_{TPN} = \{N, I\}$, *where*:
$N = \{P, T, f, m_0\}$ *is a classical Petri net*,
$I : T \to \mathbb{Q}^+ \cup \{0\} \times \mathbb{Q}^+ \cup \{0\} \cup \{\infty\}$, *where for every transition* $t \in T$, *with* $I(t) = [I_1(t), I_2(t)]$, *it holds that* $I_1(t) \leq I_2(t)$.

In this definition I is called a time function for N, where $I_1(t)$ denotes the earliest firing time for transition t, $I_2(t)$ denotes the latest firing time for t. Time counting for every transition begins when it becomes enabled. Transition will fire after time z: $I_1(t) \leq z \leq I_2(t)$, i.e., transition may fire no sooner than at $I_2(t)$ and it must fire no later than at $I_2(t)$ time. If at any time before firing transition stops being enabled, after it becomes enabled again the counting of its firing time must start from 0. The firing event is instantaneous, i.e., production and consumption of tokens takes no time. It should be noted that time values are not restricted to integers, they are in fact rational numbers. This allows more flexibility when modeling a biological system where times of processes occurring in such system can differ by orders of magnitude. A more complex time scales (e.g., a logarithmic ones) can be used in the models. Analytical techniques available for time Petri nets are not restricted by this fact, due to the possibility of full transformation of rational-value times into integer ones [14]. More complex problems, e.g., time-scale decomposition, are often discussed for stochastic Petri nets, which are mathematically similar to the discussed type of Petri nets with time [25].

A state of a time Petri net is more complex than in a classical net because it must also contain data about transitions' timers counting towards z value. For TPN its state is given as pair $z = (m, h)$, where $m : P \to \mathbb{N}$ and $h : T \to \mathbb{Q}^+ \cup \{0\} \cup \{\#\}$. To fully describe a given state of TPN, a distribution of tokens in places must be stored in the same way as it is for the classical Petri net (function m). Additionally, a state of TPN must also contain the value of every transition internal counter (function h). h assigns values from interval $[0, z]$, if the transition is enabled and symbol # otherwise [14]. It must be noted that states of TPN change not only when a transition fires (and therefore changes the numbers of tokens in places), but also when time elapses.

3. Time Petri net analysis

It should be noted that the approaches basing on, e.g., t-invariant analysis of a classical Petri net based models can still be used with time Petri nets described in this paper. It is possible because the structure of a Petri net with time remains the same as in the classical one. Such a structure is often called a net skeleton [14] and it is equivalent to the classical Petri net from Definition 1. The additional time data about subprocesses delay or duration allows a creation of more precise models of a studied biological system [15], for which a new analytical approaches explained further become available.

3.1. t-invariant analysis. The verification of the possibility to execute all transitions from a given t-invariant support is one of the relevant problems in time Petri nets analysis. Calculating t-invariants is performed in the same way as in the case of classical Petri nets. However, checking whether transitions within a support may run a specific number of times is difficult due to time limits $I_1(t)$ and $I_2(t)$. Additionally, it is important that the given verification should be available for unbounded time Petri nets or in general without the necessity to analyze state space of such a net, due to the fact that most of time Petri nets constructed to model biological systems are unbounded [14].

In the literature it can be found a mathematical approach for solving this problem based on a set of inequalities gathered from time functions $I(t)$ [14]. During our analysis a simple TPN

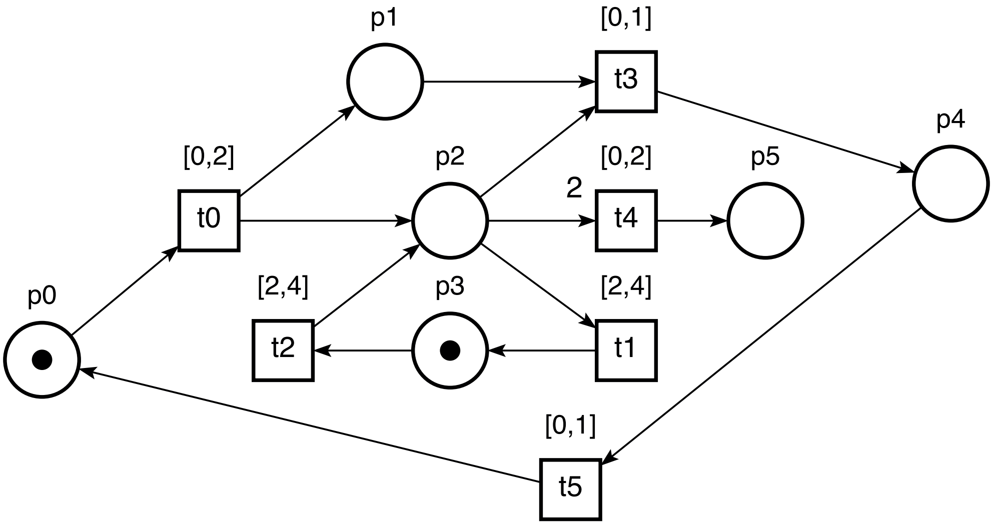

Fig. 1. Time Petri net with two t-invariants

will be presented (Fig. 1) and verified using described solution. As a next step, the expanded net (Fig. 2) with additional transitions will be analyzed. Based on this TPN, an example will be given, to confirm that a more complex verification is required. To fully determine whether selected transitions may be executed, an alternative method for solving the analogous problem will be presented.

A special notation required for further analysis is introduced in the aforementioned literature. Let $\sigma = t_{j_1}, \ldots, t_{j_w}$ be sequence of transitions and let $z_i \in \mathbb{Q}^+$ with $i = 1, \ldots, w - 1$ be the time which elapsed between the firing of transitions t_{j_i} and $t_{j_{i+1}}$. Then z_0 denotes the time from the activation of transition t_{j_1} to its execution and z_w denotes the possible time which may elapse after the last transition execution. The sequence of w transitions executed in time $z = z_0 + z_1 + \ldots + z_w$ is called a *run* of σ: $\sigma(z) = z_0 t_{j_1} z_1 \ldots z_{w-1} t_{j_w} z_w$ [14]. Additionally, for each sequence there is defined a set of conditions $B_{\sigma(z)}$ containing inequalities which describe the constraints for each variable $z_i (i = 0, \ldots, w)$ in $\sigma(z)$. This constraints are build upon time functions $I(t_j)$ (Def. 3), associated with each transition in TPN. In this paper we consider transition sequences which are build from t-invariants supports. The possibility of their execution may be verified by solving the set of inequalities $B_{\sigma(z)}$. The main objective of this analysis is a verification whether a given transition sequence σ may be executed before execution of any other enabled transitions outside the sequence.

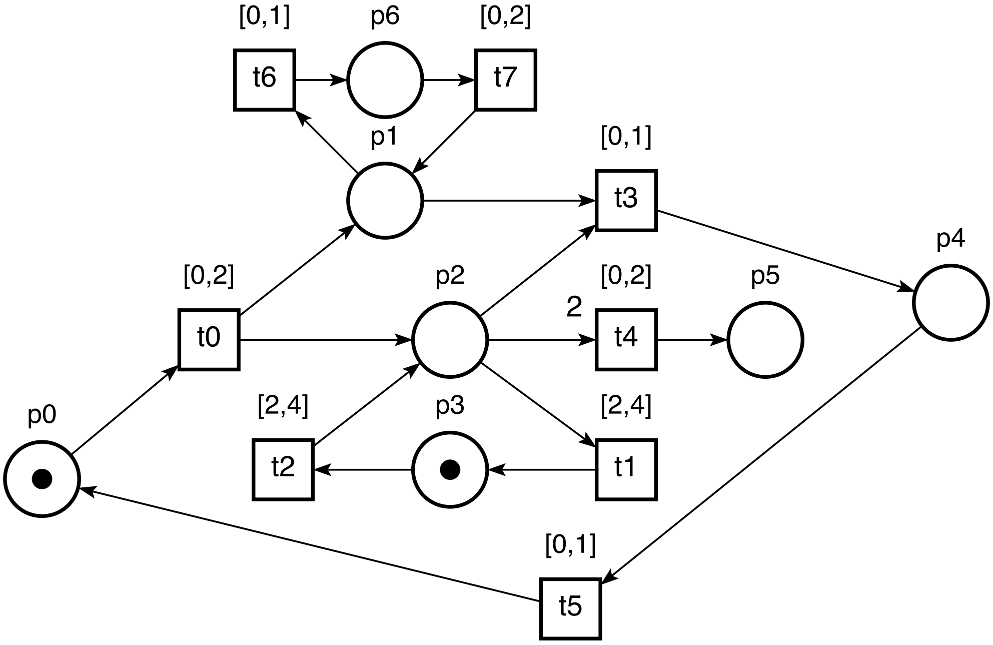

Fig. 2. Time Petri net with three t-invariants

In the following analysis this notation will be slightly modified. The finishing time z_w after the last transition execution will be ignored, because it is not affecting the verification process.

To visualize the described approach an analysis of the TPN presented in Fig. 1 will be given (cf. [26]). This network contains two t-invariants:

$x_1 = (1, 0, 0, 1, 0, 1)$ with support $s(x_1) = \{t_0, t_3, t_5\}$
$x_2 = (0, 1, 1, 0, 0, 0)$ with support $s(x_2) = \{t_1, t_2\}$
(t-invariants can be calculated using various tools, among others INA – Integrated Net Analyzer [27]).

Based on the supports of t-invariants x_1 and x_2 two transition sequences are created, which may be executed in time z:

$\sigma_1(z) = (z_0, t_0, z_1, t_3, z_2, t_5)$
$\sigma_2(z) = (z_0, t_2, z_1, t_1)$
Based on an initial marking of the network, in the second sequence transition t_2 is placed before t_1. For the sequence $\sigma_1(z)$, the following set of conditions is obtained:

$$B_{\sigma_1(z)} = \left\{ \begin{matrix} 0 \le z_0 \le 2, \; z_0 + z_1 \le 4, \\ 0 \le z_1 \le 1, \; z_0 + z_1 + z_2 \le 4, \\ 0 \le z_2 \le 1 \end{matrix} \right\}$$

It is easy to show that there exists a solution to the above set of inequalities in the set of rational numbers. Constraints presented in the inequalities are narrowed to ≤ 4, because of the latest firing time for transition t_2. This transition is not present in the sequence σ_1, but because of the token in place p_3 it is enabled. As a result it has to be taken into consideration, because its execution would change the state of the network. The aforementioned analysis confirms the possibility of execution of transitions from the first t-invariant support.

For sequence σ_2, it is easy to confirm that for the given net state it is not executable. The set of constraints is as follows:

$$B_{\sigma_2(z)} = \left\{ \begin{matrix} 2 \le z_0 \le 4, \; z_0 + z_1 \le 2, \\ 2 \le z_1 \le 4 \end{matrix} \right\}$$

There is no solution in $\mathbb{Q}^+$ for this set of inequalities. This is because transition t_0 which is not in sequence σ_2 is enabled and will be executed in time $I_2(t_0) \le 2$.

In fact, execution of the transition outside the sequence does not mean that transitions from the sequence cannot be executed. Moreover, it may cause that another concurrent transition will stop being enabled. To show the complexity of the problem, the presented TPN has been modified by adding extra transitions t_6 and t_7 (Fig. 2) (cf. [26]). During the structural analysis, three t-invariants were found:

$x_1 = (1, 0, 0, 1, 0, 1, 0, 0)$
$x_2 = (0, 1, 1, 0, 0, 0, 0, 0)$
$x_3 = (0, 0, 0, 0, 0, 0, 1, 1)$
The first and the second ones have the same supports like x_1 and x_2 for the TPN from Fig. 1. Additionally, there is a completely new t-invariant x_3 with support $s(x_3) = \{t_6, t_7\}$. It may be observed that after an execution of transition t_0, there is an additional enabled transition t_6. An execution of this new transition, causes that transition t_3 is no longer enabled (when place p_1 contains no tokens). It directly affects the possibility of execution of the transitions from sequence run $\sigma_2(z)$. Because of competing transitions t_3 and t_6, it is possible that transition t_6 will use the token from place p_1. This way transition t_3 will stop being enabled. The token will be used by transition t_7 and will return to place p_1. It is possible that, the time passed during execution of transitions t_6 and t_7 will be long enough to make transition t_1 enabled and to use a token from place p_2 before transition t_3 will be executed.

According to this, to verify every run of a sequence, it is necessary to construct a set of inequalities for every state of the analyzed network. If all constraints within this set of inequalities for a given run of a sequence has a solution in $\mathbb{Q}^+$, then for sure all transitions from this sequence may be executed. Otherwise, according to the example above, it cannot be excluded.

3.2. Minimum and maximum run of sequence. Because of such restrictive approach proposed in the aforementioned literature, the above analysis has a minor usability for longer t-invariants. According to this, additional verification method is proposed [26] based on the following definitions:

Definition 4. Feasible run [14]
Run $\sigma(z)$ is called feasible from state z in N_{TPN}, if there is state z' such that $\sigma(z)$ can fire from z into z'.

Definition 5. Firing sequence [14]
Transition sequence σ is a firing transition sequence in the TPN if there is a feasible run $\sigma(z)$ in N_{TPN}.

Definition 6. Length of a run [14]
For given TPN and a feasible run $\sigma(z)$ of firing sequence σ in N_{TPN}. The length of time $l\big(\sigma(z)\big)$ of $\sigma(z)$ is the sum of all times elapsing over the course of the firing of $\sigma(z)$.

Definition 7. Minimum run [14]
Feasible run $\sigma(z)$ of σ has minimum length of time, if there is no feasible run of σ with length of time shorter than $l\big(\sigma(z)\big)$.

The notation of maximum run may be defined analogous, if the set of all lengths of feasible runs of σ has an upper bound [14]. For the current analysis minimum and maximum length of run are calculated. As an example, two competing transitions t_3 and t_1 are selected from the second TPN (Fig. 2). The last common transition which precedes both of them is t_0. In Table 1, there are generated possible transition sequences, which are started from transition t_0 and stopped at t_3 or t_1. Under

Table 1
Minimum and maximum length of run
for the analyzed transition sequences

σ	t_0, t_1	t_0, t_3	t_0, t_6, t_7, t_3
min $l\big(\sigma(z)\big)$	2	0	0
max $l\big(\sigma(z)\big)$	6	3	6

every transition sequence minimum and maximum lengths of their run are calculated.

The minimum length of run for sequence t_0, t_1 equals to 2, when the maximum length of run for sequence t_0, t_6, t_7, t_3 is greater and equals to 6. According to this analysis, it is possible to execute transition t_1, before transition t_3. Based on that kind of comparison it is possible to check whether any transition may be executed before other concurrent transitions. Analogous analysis will be used in an extension of algorithm 1.

4. Calculating the number of tokens required for transition firing

In order to investigate the possibility of firing of all transitions from a t-invariant support, the following algorithm (Alg. 1) has been proposed. As an initial parameter there is an empty result set *RES* declared. Further notation contains t-invariant support $s(x)$ selected for further analysis. The weight of an arc from place p_i to transition t_j is denoted by $w_{p_i \to t_j}$. Additionally, symbol ε will be used to denote arbitrarily small positive number. The result of the algorithm is a set of places with minimal and maximal numbers of tokens required for selected transitions execution.

Algorithm 1

```
 1: $RES := \emptyset$
 2: for all $t_j \in s(x)$ do {
 3:   for all $p_i \in {}^\bullet t_j$ do {
 4:     if $|p_i^\bullet| > 1$ then {
 5:       $S := p_i^\bullet \setminus \{t_j\}$;
 6:       for all $t_c \in S$ do {
 7:         if $I_1(t_c) \geq I_2(t_j)$ AND $m(p_i) \geq w_{p_i \to t_c}$ then {
 8:           $checkPreconditions(S)$;
 9:           $k_{min} := \sum_{c=1}^{|S|} w_{p_i \to t_c} \left\lfloor \frac{I_1(t_j) - \varepsilon}{I_2(t_c)} \right\rfloor + w_{p_i \to t_j}$;
10:           $k_{max} := \sum_{c=1}^{|S|} w_{p_i \to t_c} \left\lfloor \frac{I_2(t_j)}{I_1(t_c)} \right\rfloor + w_{p_i \to t_j}$;
11:           $RES := RES \cup \{(p_i, [k_{min}, k_{max}])\}$;
12:           break ;
13:         }
14:       }
15:     }
16:   }
17: }
```

The algorithm is finished when all transitions from a given t-invariant support will be verified (compared with concurrent transitions). Its complexity in the worst case is $O\big(|T|^2|P|\big)$.

Based on the presented algorithm, a verification of a possibility of a t-invariant execution can be done as follows:

- For each transition t_j belonging to transition sequence σ, verification is based on checking whether pre-places of t_j have more than one post-transition. Validation of the condition in step 4 tells us whether transitions belonging to a t-invariant support are competing for tokens with another transition from this t-invariant support or from any other part of the net.
- Set S contains all transitions which are competing with t_j (without transitions connected by a read arc, because they do not consume tokens). For each of them there is a condition to check that their earlies firing time is not smaller than the latest firing time of selected transition t_j. Additionally, there is a condition to check that place p_i contains enough tokens to activate concurrent transitions.
- When the conditions are met, place p_i must contain an appropriate number of tokens to execute transition t_j. This number is calculated as an interval:
  - in the most optimistic case, the minimal number of tokens k_{min} is equal to a sum of the earliest firing time of transition t_j divided by the latest firing time of each competing transition, multiplied by the weights of arcs from p_i to t_c plus the weight of the arc from p_i to t_j. If $I_1(t_j) = I_2(t_c)$ then considering the most optimistic scenario transition t_j fires before t_c, that is why an ε is subtracted from $I_1(t_j)$.
  - in the most pessimistic case, the maximal number of tokens k_{max} is equal to a sum of the latest firing time of transition t_j divided by the earliest firing time of each competing transition plus the weight of the arc from p_i to t_j. Again, the quotient values are multiplied by the weights of the corresponding arcs.
- In order to avoid dividing by zero a dedicated procedure $checkPreconditions(S)$ has been proposed with the following conditions:
  - if there exists concurrent transition with $I_2(t_c) = 0$ and $I_1(t_j) \neq 0$, then $k_{min} = \infty$ and transition t_j cannot be fired until $|{}^\bullet t_c| = 1$
  - if there exists concurrent transition with $I_1(t_c) = 0$ and $I_2(t_j) \neq 0$, then $k_{max} = \infty$
  - if $I_1(t_j) = 0$ then $k_{min} = 1$
  - if $I_2(t_j) = 0$ then $k_{max} = 1$
- Obtained interval $[k_{min}, k_{max}]$ for place p_i means that verified transition t_j from the t-invariant support:
  - surely will be executed, if there is at least k_{max} tokens in place p_i. That is because k_{max} denotes a number of tokens required to run all competing transitions in their earliest firing time with the assumption that t_j is fired in the latest firing time.
  - may be executed, if there is at least k_{min} tokens in place p_i.
  - cannot be executed, if there is less than k_{min} tokens in place p_i and additional pre-places of competing transitions do not exist in the TPN. That is because k_{min} denotes a number of tokens required to run t_j in its earliest firing time with the assumption that all the competing transitions are fired in the latest firing time.
- All places with calculated interval $[k_{min}, k_{max}]$ are added to the result set. If the algorithm finishes with an empty result set, it is a confirmation that all transitions from the verified t-invariant support will be executed. Otherwise, transitions are guaranteed to be executed when there are k_{max} tokens in every place from the result set.

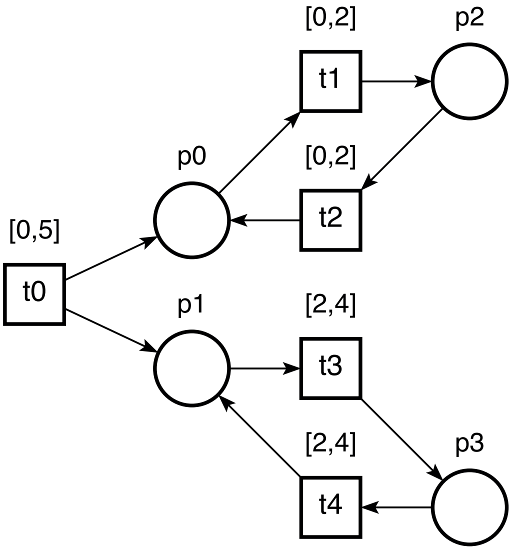

Fig. 3. Time Petri net without concurrent transitions

4.1. An example without concurrent transitions. At the beginning, proposed algorithm is adapted to verify a small time Petri net without concurrent transitions (Fig. 3). It may be observed that it contains two t-invariants:

$x_1 = (0, 1, 1, 0, 0)$ with support: $s(x_1) = \{t_1, t_2\}$ and
$x_2 = (0, 0, 0, 1, 1)$ with support: $s(x_2) = \{t_3, t_4\}$.

In the third step of Algorithm 1 there is no place found with more than one post-transition. Because there are no competing transitions in this TPN, both transitions from both t-invariant supports may be executed without any additional constraints.

4.2. An example with concurrent transitions. In Fig. 4, a time Petri net with two t-invariants is shown.

$x_1 = (0, 1, 1, 1, 0, 0, 0, 0, 1, 1)$ with support:
$s(x_1) = \{t_1, t_2, t_3, t_8, t_9\}$ and
$x_2 = (0, 0, 0, 0, 0, 0, 1, 1, 0, 0)$ with support: $s(x_2) = \{t_6, t_7\}$.

In this respect, t-invariant support $s(x_2)$ will be considered. According to Algorithm 1, there is set of transitions $S = \{t_4, t_5, t_8\}$, which are competing with transition t_6 (from $s(x_2)$) for tokens from place p_3.

Comparing firing times of concurrent transitions, the condition from step 7 is met with values $I_1(t_4) = 3$ and $I_2(t_6) = 6$. Because $I_1(t_4) > I_2(t_6)$ there are additional tokens required in place p_3 to execute transitions from $s(x_2)$:

$$k_{min} = \left\lfloor \frac{I_1(t_6) - \varepsilon}{I_2(t_4)} \right\rfloor + \left\lfloor \frac{I_1(t_6) - \varepsilon}{I_2(t_5)} \right\rfloor + \left\lfloor \frac{I_1(t_6) - \varepsilon}{I_2(t_8)} \right\rfloor + 1$$

$$k_{min} = \left\lfloor \frac{5 - \varepsilon}{4} \right\rfloor + \left\lfloor \frac{5 - \varepsilon}{7} \right\rfloor + \left\lfloor \frac{5 - \varepsilon}{2} \right\rfloor + 1$$

$$k_{min} = 1 + 0 + 2 + 1 = 4$$

$$k_{max} = \left\lfloor \frac{I_2(t_6)}{I_1(t_4)} \right\rfloor + \left\lfloor \frac{I_2(t_6)}{I_1(t_5)} \right\rfloor + \left\lfloor \frac{I_2(t_6)}{I_1(t_8)} \right\rfloor + 1$$

$$k_{max} = \left\lfloor \frac{6}{3} \right\rfloor + \left\lfloor \frac{6}{6} \right\rfloor + \left\lfloor \frac{6}{1} \right\rfloor + 1$$

$$k_{max} = 2 + 1 + 6 + 1 = 10$$

The resulted set produced by the algorithm contains one element: $(p_3, [4, 10])$. This means that transition t_6 from the second t-invariant support:

- surely will be executed, when place p_3 contains at least 10 tokens,
- might be executed when there are at least 4 tokens in place p_3.

Since competing transition t_8 has more than one pre-place, we cannot say that it is impossible to execute t_6, if there is less than 4 tokens in place p_3. This particular case will be presented in detail in the next Section describing the extension of the algorithm.

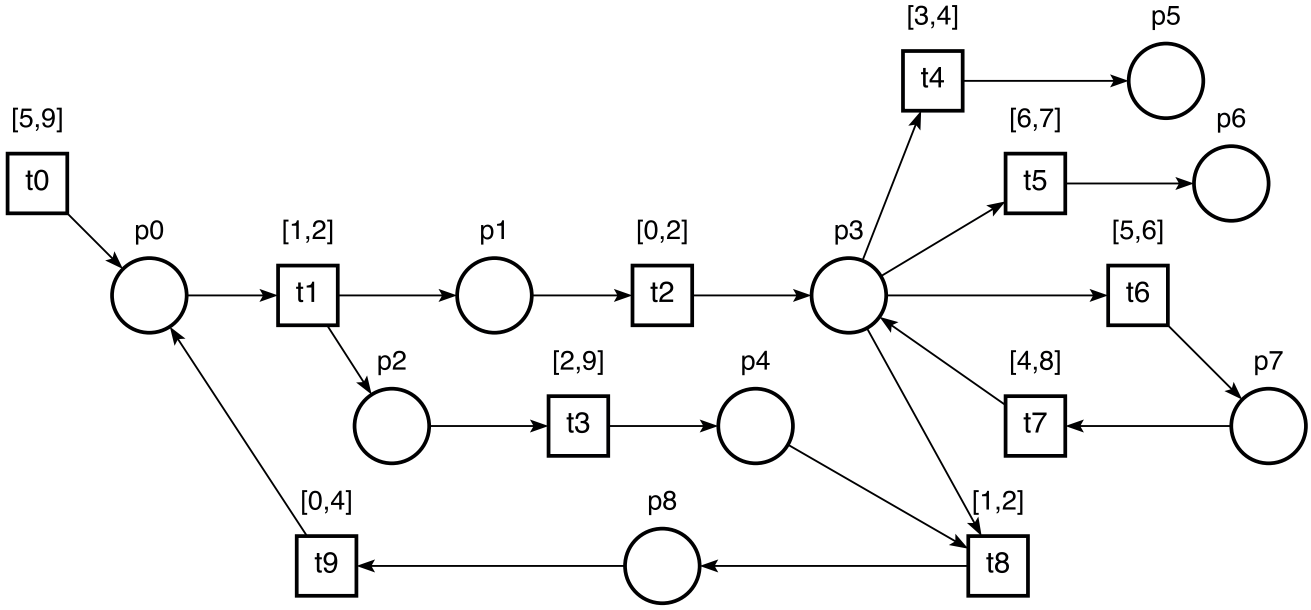

Fig. 4. Unbounded time Petri net with concurrent transitions for Algorithm 1 analysis

4.3. Algorithm extension. If in set S there exists at least one transition t_c with $|{}^{\bullet}t_c| > 1$, then it is possible to apply an additional analysis based on the approach from the previous section. Analogous to Table 1, transition sequences with minimum and maximum length of run will be generated. As a result, a decreased number of required tokens (k_{min}) may be calculated.

According to the calculation prepared for Fig. 4, there is a competing transition $t_8 \in S$ with two pre-places p_3 and p_4. When there are no tokens in place p_4, transition t_8 is not enabled and will not fire before transition t_6 from $s(x_2)$. It is possible to compare the maximum run for sequence $\sigma_1 = t_3, t_8$ and the minimum run for sequence $\sigma_2 = t_2, t_6$, because they have common initial transition t_1:

$$max\ l\big(\sigma_1(z)\big) = I_2(t_3) + I_2(t_8) = 9 + 2 = 11$$
$$min\ l\big(\sigma_2(z)\big) = I_1(t_2) + I_1(t_6) = 0 + 5 = 5.$$

Based on this calculations, it may be observed that in the most optimistic case tokens may approach place p_4 in time $I_2(t_3) = 9$. During this time, transition t_8 may not be enabled and for that time may not actively compete with transition t_6. For this particular situation it may be omitted from k_{min} calculations. Instead of that, lengths of run should be taken into account. The reduced minimal number of tokens required to execute t_6 may be calculated in the following way:

$$k_{min} = \left\lfloor \frac{I_1(t_6)}{I_2(t_4)} \right\rfloor + \left\lfloor \frac{I_1(t_6)}{I_2(t_5)} \right\rfloor + \left\lfloor \frac{min\ l(\sigma_2(z))}{max\ l(\sigma_1(z))} \right\rfloor + 1$$

$$k_{min} = \left\lfloor \frac{5}{4} \right\rfloor + \left\lfloor \frac{5}{7} \right\rfloor + \left\lfloor \frac{5}{11} \right\rfloor + 1$$

$$k_{min} = 1 + 0 + 0 + 1 = 2$$

It means that in the most optimistic case at least two tokens are required in place p_3 to execute transition t_6.

4.4. An example of virus infection TPN based model. To visualize the biological relevance of the proposed algorithm an additional Petri net has been proposed (Fig. 5). The model shows an infection of uninfected cells by virus and an immune system response. Time ranges adapted to this model are estimated to represent the real life biological behaviour. When the external infection occurs and the virus enters the cell, the replication of the virus starts. When the immune system responds to infection, cells are producing interferon and the immune system is producing immunoglobulin G (IgG). This production and accumulation is the long term response of the body, but finally it protects from infection (cf. [28]).

For the proposed model (Fig. 5), there are computed the following t-invariants:

$x_1 = (1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$x_2 = (0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 3, 0, 2, 0, 0)$
$x_3 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1)$
$x_4 = (0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)$
$x_5 = (0, 1, 1, 6, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$x_6 = (0, 3, 3, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 0)$
$x_7 = (0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 0)$
$x_8 = (0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0, 0, 1, 1, 0)$
$x_9 = (0, 6, 6, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 2, 3)$
$x_{10} = (0, 7, 7, 0, 1, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$x_{11} = (0, 7, 7, 0, 1, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0)$
$x_{12} = (0, 1, 1, 0, 1, 0, 2, 6, 0, 0, 0, 0, 2, 0, 0)$
$x_{13} = (0, 7, 7, 0, 1, 0, 0, 0, 6, 0, 0, 0, 0, 0, 3)$

with supports:

$s(x_1) = \{T0\_Cell\_death, T1\_Cell\_growth\}$
$s(x_2) = \{T6\_Immune\_system\_response,$
$T10\_IgM\_degradation, T12\_IgG\_degradation\}$

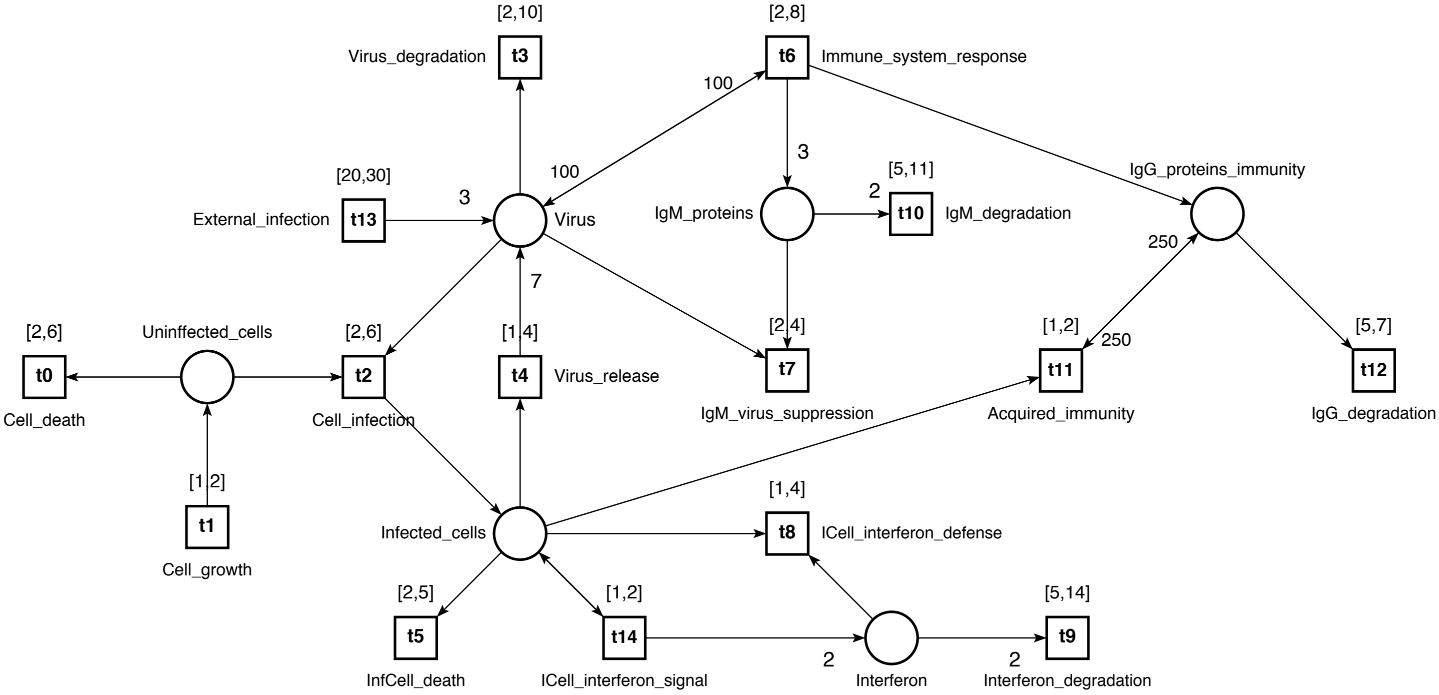

Fig. 5. Unbounded time Petri net based model of virus infection and an immune system response

$s(x_3) = \{$*T9_Interferon_degradation*, *T14_ICell_interferon_signal*$\}$

$s(x_4) = \{$*T3_Virus_degradation*, *T13_External_infection*$\}$

$s(x_5) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T3_Virus_degradation*, *T4_Virus_release*$\}$

$s(x_6) = \{$*T1*_Cell_growth, *T2*_Cell_infection, *T5*_InfCell_death, *T13*_External_infection$\}$

$s(x_7) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T11_Acquired_immunity*, *T13_External_infection*$\}$

$s(x_8) = \{$*T6_Immune_system_response*, *T7_IgM_virus_suppression*, *T12_IgG_degradation*, *T13_External_infection*$\}$

$s(x_9) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T8_ICell_interferon_defense*, *T13_External_infection*, *T14_ICell_interferon_signal*$\}$

$s(x_{10}) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T4_Virus_release*, *T5_InfCell_death*$\}$

$s(x_{11}) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T4_Virus_release*, *T11_Acquired_immunity*$\}$

$s(x_{12}) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T4_Virus_release*, *T6_Immune_system_response*, *T7_IgM_virus_suppression*, *T12_IgG_degradation*$\}$

$s(x_{13}) = \{$*T1_Cell_growth*, *T2_Cell_infection*, *T4_Virus_release*, *T8_ICell_interferon_defense*, *T14_ICell_interferon_signal*$\}$

In order to present the Algorithm 1 behaviour, an analysis of the first three t-invariants will be presented. For all of them the algorithm generates non-empty result set. For t-invariant x_1 and support $s(x_1)$ there is set S with one transition, i.e., $S = \{$*T2_Cell_infection*$\}$, which competes with transition *T0_Cell_death* (from $s(x_1)$) for tokens from place *Uninfected_cells*. It may be confirmed that in the most optimistic case only one token will be enough to execute transition t_2:

$$k_{min} = \left\lfloor \frac{I_1(t_0) - \varepsilon}{I_2(t_2)} \right\rfloor + 1 = \left\lfloor \frac{2-\varepsilon}{6} \right\rfloor + 1 = 1$$

$$k_{max} = \left\lfloor \frac{I_2(t_0)}{I_1(t_2)} \right\rfloor + 1 = \left\lfloor \frac{6}{2} \right\rfloor + 1 = 4$$

For t-invariant x_2 with support $s(x_2)$ there is set S with one transition, i.e., $S = \{$*T7_IgM_virus_suppression*$\}$, which competes with transition *T10_IgM_degradation* (from $s(x_2)$) for tokens from place *IgM_proteins*. Comparing firing times of the concurrent transitions, the condition from step 7 is met with values $I_2(t_{10}) = 11$ and $I_1(t_7) = 2$. Additionally, weight of the arc from place *IgM_proteins* to concurrent transition t_7 is equal to 1 and this place will be supplied with 3 tokens. Because $I_2(t_{10}) > I_1(t_7)$ there are additional tokens required in place *IgM_proteins* to execute transitions from $s(x_2)$:

$$k_{min} = \left\lfloor \frac{I_1(T10) - \varepsilon}{I_2(T7)} \right\rfloor + 2 = \left\lfloor \frac{5-\varepsilon}{4} \right\rfloor + 2 = 3$$

$$k_{max} = \left\lfloor \frac{I_2(T10)}{I_1(T7)} \right\rfloor + 2 = \left\lfloor \frac{11}{2} \right\rfloor + 2 = 7$$

The result set *RES* produced by the algorithm contains one element: (*IgM_proteins*, [3, 7]). This means that transition *T10_IgM_degradation* from the second t-invariant support:

- surely will be executed, when place *IgM_proteins* contains at least 7 tokens,
- might be executed when there are at least 3 tokens in place *IgM_proteins*.

Because transition *T6_Immune_system_response* generates 3 tokens for place *IgM_proteins*, execution of *T10_IgM_degradation* is always possible.

Analogously, for t-invariant x_3 support $s(x_3)$ is verified. Set S contains one transition *T8_ICell_interferon_defense*, which competes with *T9_Interferon_degradation* for tokens from place *Interferon*. Because the condition from step 7 is met $\big(I_2(t_9) > I_1(t_8)\big)$, there are additional tokens required for that place:

$$k_{min} = \left\lfloor \frac{I_1(T9) - \varepsilon}{I_2(T8)} \right\rfloor + 2 = \left\lfloor \frac{5-\varepsilon}{4} \right\rfloor + 2 = 3$$

$$k_{max} = \left\lfloor \frac{I_2(T9)}{I_1(T8)} \right\rfloor + 2 = \left\lfloor \frac{14}{1} \right\rfloor + 2 = 16$$

For t-invariant x_3, the algorithm produced set *RES* with one element: (*Interferon*, [3, 16]). This means that transition *T9_Interferon_degradation*:

- surely will be executed, when place *Interferon* contains at least 16 tokens,
- might be executed when there are at least 3 tokens in place *IgM_proteins*.
- cannot be executed, if there is less then 3 tokens in that place.

The proposed algorithm adapted to the time Petri net based model of virus infection, may return all places which require an additional number of tokens. Based on these values, it is possible to identify transitions from selected t-invariant supports in which execution will be delayed or less realistic. This gives a strong biological relevance in further model analysis.

5. Preparation and verification of time parameters

During the creation of a time Petri net based model of some complex biological system, one may encounter a problem of a selection of the proper time values for transitions. Such parameters should assure that the net will behave in accordance with the current knowledge about the modeled biological system. While searching for this knowledge in the literature, there is a possibility that the times describing the analyzed reactions are given imprecisely, or for some parts of the biological system they are not given at all. For models of some biological systems, such data can be obtained using analytical methods taking into account the structure of the net as well as some characteristics of the modeled system.

A method for establishing time parameters that follows from the net structure has been proposed in [16] for signaling pathways modeled using timed Petri net (DPN). Such a net is defined as follows:

Definition 8. Timed Petri net (DPN) [14]
Timed Petri net is a set $N_{DPN} = \{N, D\}$, *where:*
$N = \{P, T, f, m_0\}$ *is a classical Petri net,*
$D : T \to \mathbb{Q}^+$.

D is a function that assigns to every transition t_j delay value d_j being a positive rational number. In such a net it is always assumed that an enabled transition has to fire immediately. In contrast to other nets, only the consumption of tokens from pre-places of a transition is an instantaneous action. When it happens, the transition starts counting time from 0 to d_i. When the transition internal counter reaches time d_j, then the transition produces tokens in its post-places. An enabled transition that started counting time towards d_j will always produce tokens (after time d_j), no matter if it is still enabled or not after it (immediately) consumed tokens from pre-places.

In the methodology for establishing time parameters (i.e., delays in the aforementioned timed Petri net) described in [16] one feature of a system modeled is especially important, i.e., a lack of cycles. This approach is based on a system of equations and inequalities, using the so called firing frequencies for transitions. As it is stated in [16], in a timed Petri net a firing frequency f_j of a transition t_j is constrained by its delay time d_j and the maximum of firing frequency is the reciprocal of d_j. A firing frequency is a value which tells how many number of times a given transition fire in some pre-established time unit. Here we will use the notation where f_{O_a} and f_{I_b} describe respectively the firing frequency of some post-transition (t_a) and pre-transition (t_b) of place p_i. Indexes *a* and *b* will be used to help in distinguishing between post- and pre-transitions of p_i.

The whole approach is based on an assumption that for every substance represented by places, its total production (by transition firing) is equal to its consumption in a given time unit. Therefore, for every place p_i the following equation must hold:

$$\sum_{a=1}^{m_a} K_{O_a} = \sum_{b=1}^{m_b} K_{I_b}. \tag{1}$$

In equation (1), value $m_a = |p_i{}^{\bullet}|$ and $m_b = |{}^{\bullet}p_i|$. K_{O_a} denotes a number of tokens taken from p_i in an established time unit by a transition $t_a \in p_i{}^{\bullet}$, while K_{I_b} denotes a number of tokens produced in place p_i in the same time unit by some transition $t_b \in {}^{\bullet}p_i$. This equation has to hold for the maximal firing frequencies of transitions. It can be achieved in a timed Petri net, where all enabled transitions must fire immediately. Because firing frequency f_j is a reciprocal of delay value d_j, the latter one can be established on the basis of f_j.

There are two mathematical rules to determine the speed of a transition firing. Firstly, for every place p_i with only one post-transition t_O and with at least one pre-transition t_{I_b}, the maximum firing frequency f_j must satisfy the following equation:

$$\sum_{b=1}^{m_b} \beta_b \cdot f_{I_b} = \alpha \cdot f_O. \tag{2}$$

In the above equation $m_b = |{}^{\bullet}p_i|$, while α and β_b are weights of arcs (p_i, t_O) and (t_{I_b}, p_i) respectively. f_{I_b} and f_O denote the maximal firing frequencies for the m$_b$ pre-transitions and for the one post-transition t_O respectively.

The second rule applies to any place p_i for which there is a conflict of post-transitions. In such a case the transitions must have firing frequencies *f* for which:

$$\begin{cases} \sum_{a=1}^{m_a} \alpha_a \cdot f_{O_a} = \sum_{b=1}^{m_b} \beta_b \cdot f_{I_b} \\ 2 \cdot \frac{f_{Om_a}}{\alpha_{m_a}} \geq \frac{f_{O_1}}{\alpha_1} \geq \frac{f_{O_2}}{\alpha_2} \geq \ldots \geq \frac{f_{Om_a}}{\alpha_{m_a}}. \end{cases} \tag{3}$$

where α_a and β_b describe the weights of arcs (p_i, t_{O_a}) and (t_{I_b}, p_i) respectively and α_a satisfies $\alpha_1 \geq \alpha_2 \ldots \geq \alpha_{m_a}$. f_{O_a} and f_{I_b} are the maximum firing frequencies of post- and pre-transitions respectively.

The presented approach can help in establishing delays for a timed Petri net. The drawback of the this approach is that the delays computed in such a way for the conflicted transitions will always be the same with respect to the values of weights of the given arcs, i.e., different delays for such transitions are possible only if there are different α_j values (i.e., the weights). Therefore, two transitions in conflict, representing hypothetically very different reactions, will have the same delays if the weights of their arcs are equal. This is obviously an issue when modeling biological systems.

For this reason in [29] an extension of the method has been proposed, allowing the assignment of different delays for the conflicted transitions. Another important addition to the described approach is that one can assign a time range for the reactions, allowing one to use time Petri net (TPN) based models. Because of the rather complex nature of such addition, only the very basis of it will be presented in this section. The new approach is based on the so called retention-free net, in which the flow of tokens in every pre-established time unit for a given place p_x is governed by the following inequality:

$$\sum_{b=1}^{m_b} K_{I_b} \leq \sum_{a=1}^{m_a} K_{O_a}. \tag{4}$$

Variables in the above inequality are the same as for the already described equation (1), and the difference between the two is quite obvious. For a net where the following inequity holds (i.e., retention-free net introduced [29]) in a given time unit for every place p_i the number of tokens produced in p_i by its pre-transitions is never greater than the number of tokens taken from p_i by its post-transition in the same time unit. In other words, the net designed using this methodology in all its places produces no more tokens than can be consumed at a given time.

Both methods offer new possibilities. In a case when one lacks the data about the reaction times, they can be drawn directly from the net structure, assuming there exist some equilibrium in the biological system that can be represented by the balance of the flow of tokens in the net. In [16] and [29] the authors proved the usability of their methods for signal transduction in apoptotic pathway and the signaling pathway for interleukin-1, respectively. The calculated times and delays allows the steady flow of tokens resulting in keeping the whole system balanced. However, the problem with cycles within the net making the calculation of the *exact* time values impossible remains. The authors acknowledge the existence of such an issue [29], studying of which can further extend the whole proposed methodology.

6. Other types of time Petri nets

Theoretically one can try to combine (to some extent) both types of Petri nets with time (TPN and DPN) into a net where firing a transition is defined by a range (I from Definition 3), but when a transition fires it only consumes tokens and starts counting from 0 to d_i. Such a net behaves similar to DPN described in Definition 8, except the fact that enabled transition will not fire immediately, but after time z in range I.

In the nets discussed so far time is a parameter assigned to the transitions. There is however another type of Petri net with time, where the time is assigned to places. In [17] a deterministic interval timed places Petri net (DITPPN) has been given. Definition 9 describes such a net:

Definition 9. Deterministic interval timed places Petri net (DITPPN)
Such a net is a set $N_{DITPPN} = \{N, F\}$, *where*:
$N = \{P, T, f, m_0\}$ *is a classical Petri net*,
$F : P \rightarrow [\mathbb{Q}^+ \cup \{0\}, \mathbb{Q}^+ \cup \{0\}]$

Transition firing remains the same as in the classical Petri nets. However, enabling transition is governed by different rules which must be explained. Function F assigns two time values to places. The first one denotes a minimum time for any token in a given place that has to pass, before that token can be considered when checking whether a post-transition is enabled or not. A transition is considered enabled if in all of its pre-places there are enough tokens (i.e., their number is greater or equal to the arc weight) that have been present in these places for at least a minimum time given by F. On the other hand, a token cannot remain in a place for more time than given by the maximum value assigned by F. A concept of dead token is introduced, to denote tokens that cannot by used anymore to enable transition.

At the end of this Section a potential usage for the presented Petri nets with time parameters in the area of systems biology should be discussed. Using a net from Definition 3 can help in the modeling of system subprocesses, where reactions durations are neglected, but their occurrence is bounded by a minimal and maximal time, so it can be assigned to the transition as a range. On the other hand, if a precise duration of the reaction is known, a DPN time net (Definition 8) can be used. From the biological perspective, a net described in Definition 9 is a very precise tool to model such systems. Very often chemicals compound require some time to "grow up" in order to become functional. Also, biological compounds have a lifespan, which can be defined by the maximal time a token may reside in a place before it become unusable. In such a way a degradation of the biological compounds can be modeled.

7. Conclusions

The time of biological reactions has an a significant influence on the whole behaviour of the modeled system. Petri nets with time allow detailed and thorough modeling and analysis of the modeled system. There is more than one definition of such Petri nets, what gives researchers a possibility to choose a proper type of the net for the modeling of the studied biological system. The analysis of a Petri net often bases on a calculated set of t-invariants and time parameters allow the enhancement of this approach. Such an analysis can be enhanced, e.g., by calculating minimal and maximal times necessary to fire all the transitions from a t-invariant support. Time constraints may require that transitions fire after some time, when they are enabled. This in turn, may lead to the situation, when some transitions will be unable to fire, because of the limited number of tokens in their pre-places and the fast-firing competing transitions that will quickly deplete tokens from such places. The presented algorithm provides the ability to calculate the necessary number of tokens for places, to ensure that all competing transitions will (or will not) have a chance to fire. From this, the possibility of t-invariant execution can be estimated. Another problem described in the paper is the assignment of time parameters, i.e., the data concerning basic components of the modeled system may be inadequate to fully and precisely assign time values to transitions. To solve this problem new methods from recent publications have been described, which can be used to obtain time values from the net structure or to verify already assigned times resulting from experimental knowledge.

Acknowledgements. This work was partially supported by the National Science Centre (Poland) grant No. 2012/07/B/ST6/01537.

References

[1] Z. Szallasi, J. Stelling, and V. Periwal, *System Modeling in Cellural Biology. From Concepts to Nuts and Bolts*, MIT Press Cambridge, Massachusetts, 2006.

[2] P. Formanowicz, "On the border between biology, mathematics and computer science", *BioTechnologia* 92, 217–220 (2011).

[3] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, and R. Herwig, *Systems Biology: A Textbook*, Wiley-VCH, Weinheim, 2009.

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

[5] C. Chaouiya, "Petri net modelling of biological networks", *Briefings in Bioinformatics*, 8(4), 210–219 (2007).

[6] I. Koch, W. Reisig, and F. Schreiber (Eds.), *Modeling in Systems Biology. The Petri Net Approach*, Springer-Verlag, London, 2011.

[7] P. Baldan, N. Cocco, A. Marin and M. Simeoni, "Petri nets for modelling metabolic pathways: a survey", *Natural Computing*, 9(4), 955–989 (2010).

[8] A. Sackmann, M. Heiner, and I. Koch, "Application of Petri net based analysis techniques to signal transduction pathway", *BMC Bioinformatics*, 7, 482 (2006).

[9] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B.H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch, "Modularization of biochemical networks based on classification of Petri net t-invariants", *BMC Bioinformatics* 9, 90 (2008).

[10] D. Formanowicz, M. Radom, P. Zawierucha, and P. Formanowicz, "Petri net-based approach to modeling and analysis of selected aspects of the molecular regulation of angiogenesis", *PLOS ONE* 12(3), e0173020 (2017).

[11] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*, Springer Verlag, Berlin Heidelberg, 2010.

[12] L. Popova, "On time Petri nets", *Journal of Information Processing and Cybernetics* 27(4), 227–244 (1991).

[13] W.M. Zuberek, "Timed Petri nets definitions, properties, and applications", *Microelectronics Reliability*, 31(4), 627–644 (1991).

[14] L. Popova, *Time and Petri Nets*, Springer Verlag, Berlin Heidelberg, 2013.

[15] L. Popova and M. Heiner, and I. Koch, "Time Petri nets for modelling and analysis of biochemical networks", *Fundamenta Informaticae*, 67, 149–162 (2005).

[16] C. Li, Q.-W. Ge, M. Nakata, H. Matsuno, and S. Miyano S, "Modelling and simulation of signal transduction in an apoptosis pathway by using timed Petri nets", *Journal of Biosciences* 32 (1), 113–127 (2007).

[17] J. Błażewicz, D. Formanowicz, P. Formanowicz, A. Sackmann, and M. Sajkowski, "Modeling the process of human body iron homeostasis using a variant of timed Petri nets", *Discrete Applied Methematics* 157, 2221–2231 (2009).

[18] L. Popova and E. Pelz, "Studying steady states in biochemical reaction systems by time Petri nets", *Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets*, 71–86 (2011).

[19] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology", *Lecture Notes in Computer Science*, 5016, 215–264 (2008).

[20] M. Heiner, "Understanding network behavior by structured representations of transition invariants", *Algorithmic Bioprocesses*, 367–389 (2009).

[21] D. Formanowicz, A. Kozak, T. Głowacki, M. Radom, and P. Formanowicz, "Hemojuvelin-hepcidin axis modeled and analyzed using Petri nets", *Journal of Biomedical Informatics* 46(6), 1030–1043 (2013).

[22] S. Schuster and C. Hilgetag, "On elementary flux modes in biochemical reaction systems at steady state", *Journal of Biological Systems* 2, 165–182 (1994).

[23] I. Zevedei-Oancea and S. Schuster. "Topological analysis of metabolic networks based on Petri net theory", *In Silico Biology* 3(3), 323–245 (2003).

[24] J. Behre, L.F. de Figueiredo, S. Schuster, and C. Kaleta, "Detecting structural invariants in biological reaction networks", *Methods in Molecular Biology* 804, 377–407 (2012).

[25] J. Wang, *Timed Petri Nets*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1998.

[26] M. Radom, J. Olszak, and P. Formanowicz, "Metody analizy modeli systemów biologicznych opartych na czasowych sieciach Petriego", In: A. Świerniak, J. Krystek (eds.), Automatyzacja procesów dyskretnych. Teoria i zastosowania. Vol. II, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 189–203 (2016).

[27] INA – Integrated Net Analyzer, http://www2.informatik.huberlin.de/starke/ina.html.

[28] P.J. Delves, S.J. Martin, D.R. Burton, and I.M. Roitt, *Roitt's Essential Immunology*, 13th Edition. Wiley-Blackwell, Chichester 2017.

[29] Y. Miwa, C. Li, Q.-W. Ge, H. Matsuno, and S. Miyano, "On determining firing delay time of transitions for Petri net based signaling pathways by introducing stochastic decision rules", *In Silico Biology* 10(1), 49–66 (2010).