Towards Discrete-Time Verification of Time Petri Nets with Dense-Time Semantics*

Agata Janowska[1], Wojciech Penczek[2], Agata Półrola[3], and Andrzej Zbrzezny[4]

[1] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
janowska@mimuw.edu.pl
[2] Institute of Computer Science, PAS, Ordona 21, 01-237 Warsaw, Poland
penczek@ipipan.waw.pl
[3] University of Łódź, FMCS, Banacha 22, 90-238 Łódź, Poland
polrola@math.uni.lodz.pl
[4] Jan Długosz University, IMCS, Armii Krajowej 13/15, 42-200 Częstochowa, Poland
a.zbrzezny@ajd.czest.pl

Abstract. Verification of timed systems is an important subject of research, and one of its crucial aspects is the efficiency of the methods developed. Extending the result of Popova which states that integer time steps are sufficient to test reachability properties of time Petri nets [2, 3], in our work we prove that the discrete-time semantics is also sufficient to verify $\mathrm{ECTL}^*_{-\mathrm{X}}$ properties of TPNs with the dense semantics. To show that considering this semantics instead of the dense one is profitable, we compare the results for SAT-based bounded model checking of $\mathrm{ECTL}_{-\mathrm{X}}$ properties and the class of distributed time Petri nets.

1 Introduction

Verification of time-dependent systems is an important subject of research. The crucial problem to deal with is the state explosion: the state spaces of these systems are usually very large due to infinity of the dense time domain, and are likely to grow exponentially in the number of concurrent components of the system. This influences strongly the efficiency of the model checking methods.

The papers of Popova [2, 3] show that in the case of checking reachability for systems modelled by time Petri nets (i.e., while testing whether a marking of a net is reachable) one can use discrete (integer) time steps instead of real-valued ones. This reduces the state space to be searched. The aim of our work is to investigate whether the result of Popova can be extended, i.e., whether the discrete-time semantics can replace the dense-time one also while verifying a wider class of properties of dense-time Petri net systems. In this paper we present our preliminary result, i.e., prove that the discrete-time model can be used instead of the dense-time one while verifying $\mathrm{ECTL}^*_{-\mathrm{X}}$ properties. To show that such an approach can be profitable we perform some experiments, using

* Partly supported by the Polish Ministry of Science and Higher Education under the grant No. N N206 258035.

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 215-228

an implementation for SAT-based bounded model checking of ECTL$_{-\mathrm{X}}$ and the class of distributed time Petri nets [1].

The rest of the paper is organised as follows: Sec. 2 introduces time Petri nets and their dense and discrete models. Sec. 3 presents the logic ECTL$^*_{-\mathrm{X}}$. Sec. 4 deals with the theoretical considerations, while Sec. 5 presents the experimental results. Sec. 6 contains final remarks and sketches directions of the further work.

2 Time Petri Nets

Let $\mathbb{N}$ be the set of natural numbers (including zero), and $\mathbb{R}$ ($\mathbb{R}_+$) be the set of (nonnegative) reals. Time Petri nets are defined as follows:

Definition 1. *A time Petri net (TPN, for short) is a six-element tuple* $\mathcal{N} = (P, T, F, Eft, Lft, m^0)$*, where* $P = \{p_1, \ldots, p_{n_P}\}$ *is a finite set of* places, $T = \{t_1, \ldots, t_{n_T}\}$ *is a finite set of* transitions, $F \subseteq (P \times T) \cup (T \times P)$ *is the* flow relation, $Eft, Lft : T \to \mathbb{N}$ *are functions describing the* earliest *and the* latest firing time *of the transition, where for each* $t \in T$ *we have* $Eft(t) \leq Lft(t)$*, and* $m^0 \subseteq P$ *is the* initial marking *of* $\mathcal{N}$*.*

For a transition $t \in T$ we define its *preset* $\bullet t = \{p \in P \mid (p,t) \in F\}$ and *postset* $t\bullet = \{p \in P \mid (t,p) \in F\}$, and consider only the nets such that for each transition the preset and the postset are nonempty. We need also the following notations and definitions:

- a *marking* of $\mathcal{N}$ is any subset $m \subseteq P$,
- a transition $t \in T$ is *enabled* at m ($m[t\rangle$ for short) if $\bullet t \subseteq m$ and $t\bullet \cap (m \setminus \bullet t) = \emptyset$; and *leads from* m *to* m', if it is enabled at m, and $m' = (m \setminus \bullet t) \cup t\bullet$. The marking m' is denoted by $m[t\rangle$ as well, if this does not lead to misunderstanding.
- $en(m) = \{t \in T \mid m[t\rangle\}$;
- for $t \in en(m)$, $newly\_en(m,t) = \{u \in T \mid u \in en(m[t\rangle) \ \wedge \ (t\bullet \cap \bullet u \neq \emptyset \vee u\bullet \cap \bullet t \neq \emptyset)\}$.

Concerning the behaviour of time Petri nets, it is possible to consider a dense-time semantics, i.e. the one in which the time steps can be of an arbitrary (nonnegative) real-valued length, and the discrete one which considers integer time passings only. Below we define both of them.

2.1 Dense-Time Semantics

In the dense-time semantics (the *dense semantics* in short) a *concrete state* σ of a net $\mathcal{N}$ is defined as a pair $(m, clock)$, where m is a marking, and $clock : T \to \mathbb{R}_+$ is a function which for each transition $t \in en(m)$ gives the time elapsed since t became enabled most recently, and assigns zero to other transitions. Given a state $(m, clock)$ and $\delta \in \mathbb{R}_+$, denote by $clock + \delta$ the function defined by $(clock + \delta)(t) = clock(t) + \delta$ for each $t \in en(m)$, and $(clock + \delta)(t) = 0$ otherwise. By $(m, clock) + \delta$ we denote $(m, clock + \delta)$. The *dense concrete state space* of $\mathcal{N}$

is a structure $(T \cup \mathbb{R}_+, \Sigma, \sigma^0, \rightarrow_r)$, where Σ is the set of all the concrete states of $\mathcal{N}$, $\sigma^0 = (m^0, clock^0)$ with $clock^0(t) = 0$ for each $t \in T$ is the initial state of $\mathcal{N}$, and $\rightarrow_r \subseteq \Sigma \times (T \cup \mathbb{R}_+) \times \Sigma$ is a timed consecution relation defined by:

- for $\delta \in \mathbb{R}_+$, $(m, clock) \xrightarrow{\delta}_r (m, clock + \delta)$ iff $(clock + \delta)(t) \leq Lft(t)$ for all $t \in en(m)$ (*time successor*),
- for $t \in T$, $(m, clock) \xrightarrow{t}_r (m', clock')$ iff $t \in en(m)$, $Eft(t) \leq clock(t) \leq Lft(t)$, $m' = m[t\rangle$, and for all $u \in T$ we have $clock'(u) = 0$ for $u \in newly\_en(m, t)$ and $clock'(u) = clock(u)$ otherwise (*action successor*).

Notice that firing of a transition takes no time.

Given a set of propositional variables PV, we introduce a valuation function $V : \Sigma \rightarrow 2^{PV}$ which assign the same propositions to the states with the same markings. We assume the set PV to be such that each $q \in PV$ corresponds to exactly one $p \in P$, and use the same names for the propositions and the places. The function V is then defined by $p \in V(\sigma)$ iff $p \in m$ for each $\sigma = (m, \cdot)$. The structure $M_r(\mathcal{N}) = (T \cup \mathbb{R}_+, \Sigma, \sigma^0, \rightarrow_r, V)$ is a *dense concrete model* of $\mathcal{N}$.

A *dense σ-run* of TPN $\mathcal{N}$ is a (maximal) sequence of states: $\sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \sigma_2 \xrightarrow{a_2}_r \ldots$, where $\sigma_0 = \sigma \in \Sigma$ and $a_i \in T \cup \mathbb{R}_+$ for each $i \geq 0$. A state σ is reachable in $M_r(\mathcal{N})$ if there is a dense σ^0-run $\sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \sigma_2 \xrightarrow{a_2}_r \ldots$ such that $\sigma = \sigma_i$ for some $i \in \mathbb{N}$.

2.2 Discrete-Time Semantics

Alternatively, one can consider integer time passings only. In such a *discrete-time semantics* (*discrete semantics* in short) a (*discrete*) *concrete state* σ_n of a net $\mathcal{N}$ is a pair $(m, clock_n)$, where m is a marking, and $clock_n : T \rightarrow \mathbb{N}$ is a function which for each transition $t \in en(m)$ gives the time elapsed since t became enabled most recently, and assigns zero to the other transitions. Given a state $(m, clock_n)$ and $\delta \in \mathbb{N}$, we define $clock_n + \delta$ and $(m, clock_n) + \delta$ analogously as in the dense-time case. The *discrete concrete state space* of $\mathcal{N}$ is a structure $(T \cup \mathbb{N}, \Sigma_n, {\sigma_n}^0, \rightarrow_n)$, where Σ_n is the set of all the discrete concrete states of $\mathcal{N}$, ${\sigma_n}^0 = (m^0, {clock_n}^0)$ with ${clock_n}^0(t) = 0$ for each $t \in T$ is the initial state of $\mathcal{N}$, and $\rightarrow_n \subseteq \Sigma_n \times (T \cup \mathbb{N}) \times \Sigma_n$ is a timed consecution relation defined by:

- for $\delta \in \mathbb{N}$, $(m, clock_n) \xrightarrow{\delta}_n (m, clock_n + \delta)$ iff $(clock_n + \delta)(t) \leq Lft(t)$ for all $t \in en(m)$ (*time successor*),
- for $t \in T$, $(m, clock_n) \xrightarrow{t}_n (m', {clock_n}')$ iff $t \in en(m)$, $Eft(t) \leq clock_n(t) \leq Lft(t)$, $m' = m[t\rangle$, and for all $u \in T$ we have ${clock_n}'(u) = 0$ for $u \in newly\_en(m, t)$ and ${clock_n}'(u) = clock_n(u)$ otherwise (*action successor*).

Again, firing of a transition takes no time.

Given a set of propositional variables PV, we introduce valuation function $V_n : \Sigma_n \rightarrow 2^{PV}$ which assign the same propositions to the states with the same markings. Similarly as in the dense case, we assume the set PV to be such that each $q \in PV$ corresponds to exactly one $p \in P$, and use the same names for the

propositions and the places. The function V_n is then defined by $p \in V_n(\sigma_n)$ iff $p \in m$ for each $\sigma_n = (m, \cdot)$. The structure $M_n(\mathcal{N}) = (T \cup \mathbb{N}, \Sigma_n, {\sigma_n}^0, \rightarrow_n, V_n)$ is a *discrete concrete model* of $\mathcal{N}$.

A *discrete* σ_n*-run* of TPN $\mathcal{N}$ is a (maximal) sequence of states: $\sigma_{n0} \stackrel{a_0}{\rightarrow}_n \sigma_{n1} \stackrel{a_1}{\rightarrow}_n \sigma_{n2} \stackrel{a_2}{\rightarrow}_n \ldots$, where $\sigma_{n0} = \sigma_n \in \Sigma_n$ and $a_i \in T \cup \mathbb{N}$ for each $i \geq 0$. A state σ_n is *reachable* in $M_n(\mathcal{N})$ iff there is a ${\sigma_n}^0$-run of $\mathcal{N}$ $\sigma_{n0} \stackrel{a_0}{\rightarrow}_r \sigma_{n1} \stackrel{a_1}{\rightarrow}_r \sigma_{n2} \stackrel{a_2}{\rightarrow}_r \ldots$ such that $\sigma_n = \sigma_{ni}$ for some $i \in \mathbb{N}$.

3 The Logic ECTL$^*_{-\mathrm{X}}$

In our work we deal with verification of ECTL$^*_{-\mathrm{X}}$ properties of time Petri nets. The logic ECTL$^*_{-\mathrm{X}}$ is a sublogic of the standard branching time logic CTL*, defined as a restriction to its existential part without the next-step operator X. Below, we define the logic of our interest.

3.1 Syntax and Sublogics of ECTL$^*_{-\mathrm{X}}$

Let $PV = \{\wp_1, \wp_2 \ldots\}$ be a set of propositional variables such that $\{true, false\} \subseteq PV$. The language of ECTL$^*_{-\mathrm{X}}$ is given as the set of all the state formulas φ_s (interpreted at states of a model), defined using path formulas φ_p (interpreted along paths of a model), by the following grammar:

$$\varphi_s := \wp \mid \neg\wp \mid \varphi_s \wedge \varphi_s \mid \varphi_s \vee \varphi_s \mid \mathrm{E}\varphi_p$$
$$\varphi_p := \varphi_s \mid \varphi_p \wedge \varphi_p \mid \varphi_p \vee \varphi_p \mid \mathrm{G}\varphi_p \mid \varphi_p \mathrm{U} \varphi_p.$$

In the above $\wp \in PV$, E ('there Exists a path') is a path quantifier, whereas U ('Until') and G ('globally') are state operators. Intuitively, the formula $\mathrm{G}\varphi_p$ specifies that φ_p holds continuously along the path, whereas $\mathrm{U}(\varphi_p, \psi_p)$ expresses that ψ_p eventually occurs and that φ_p holds continuously until then. We define two derived operators: $\mathrm{F}\varphi_p \stackrel{def}{=} true \mathrm{U} \varphi_p$, and $\varphi_p \mathrm{R} \psi_p \stackrel{def}{=} (\psi_p \mathrm{U} (\varphi_p \wedge \psi_p)) \vee \mathrm{G}\psi_p$. Intuitively, the formula $\mathrm{F}\varphi_p$ specifies that φ_p occurs in some state of the path ('Finally'), whereas $\varphi_p \mathrm{R} \psi_p$ ('Release') specifies that either ψ_p holds always or it is released when φ_p eventually occurs.

One can define several sublogics of ECTL$^*_{-\mathrm{X}}$. One of them we are interested in in our work is the logic ECTL$_{-\mathrm{X}}$, which is a fragment of ECTL$^*_{-\mathrm{X}}$ in which the syntax of path formulas is restricted such that each state operator must be preceded by a path quantifer (i.e., the modalities E, G, U can appear in the combinations EU and EG only).

3.2 Semantics of ECTL$^*_{-\mathrm{X}}$

Let PV be a set of propositions. A *model* for ECTL$^*_{-\mathrm{X}}$ is a a tuple $M = (L, S, s^0, \rightarrow, V)$, where L is a set of labels, S is a set of states, $s^0 \in S$ is the initial state, $\rightarrow \subseteq S \times L \times S$ is a total successor relation[1], and $V : S \longrightarrow 2^{PV}$

[1] Totality means that $(\forall s \in S)(\exists s' \in S)\ s \rightarrow s'$.

is a valuation function. For $s, s' \in S$ the notation $s \to s'$ means that there is $l \in L$ such that $s \xrightarrow{l} s'$. Moreover, for $s_0 \in S$ a *path* $\pi = (s_0, s_1, \ldots)$ is an infinite sequence of states in S starting at s_0, where $s_i \to s_{i+1}$ for all $i \geq 0$, $\pi_i = (s_i, s_{i+1}, \ldots)$ is the i-th suffix of π, and $\pi(i) = s_i$.

Given a model M, a state s, and a path π of M, by $M, s \models \varphi$ ($M, \pi \models \varphi$) we mean that φ holds in the state s (along the path π, respectively) of the model M. The model is sometimes omitted if it is clear from the context. The relation $\models$ is defined inductively as follows:

$$
\begin{array}{lll}
M, s \models \wp & \text{iff} & \wp \in V(s), \text{ for } \wp \in PV, \\
M, s \models \neg\wp & \text{iff} & M, s \not\models \wp, \text{ for } \wp \in PV, \\
M, x \models \varphi \wedge \psi & \text{iff} & M, x \models \varphi \text{ and } M, x \models \psi, \text{ for } x \in \{s, \pi\}, \\
M, x \models \varphi \vee \psi & \text{iff} & M, x \models \varphi \text{ or } M, x \models \psi, \text{ for } x \in \{s, \pi\}, \\
M, s \models \mathrm{E}\varphi & \text{iff} & M, \pi \models \varphi \text{ for some path } \pi \text{ starting at } s, \\
M, \pi \models \varphi & \text{iff} & M, \pi(0) \models \varphi, \text{ for a state formula } \varphi, \\
M, \pi \models \mathrm{G}\varphi & \text{iff} & (\forall j \geq 0)(M, \pi_j \models \varphi), \\
M, \pi \models \varphi \mathrm{U} \psi & \text{iff} & (\exists j \geq 0)\big(M, \pi_j \models \psi \text{ and } (\forall 0 \leq i < j)\ M, \pi_i \models \varphi\big),
\end{array}
$$

Moreover, we assume $M \models \varphi$ iff $M, s^0 \models \varphi$, where s^0 is the initial state of M.

4 Discrete- vs. Dense-Time Verification for ECTL$^*_{-\mathrm{X}}$

It is easy to see that both the model $M_r(\mathcal{N})$ and $M_n(\mathcal{N})$ can be used in ECTL$^*_{-\mathrm{X}}$ verification. However, it is also not difficult to see that the latter model is smaller and less prone to the state explosion problem. The aim of our work is then to show that this model is sufficient to check ECTL$^*_{-\mathrm{X}}$ properties of time Petri nets even if we want to consider the time steps of real-valued lengths (i.e., to deal with the dense-time case). We make use of the results of Popova [2, 3].

Consider the dense concrete model $M_r(\mathcal{N}) = (T \cup \mathbb{R}_+, \Sigma, \sigma^0, \to_r, V)$ of a TPN $\mathcal{N}$. A state $\sigma = (m, clock) \in \Sigma$ is called an *integer-state* if $clock(t) \in \mathbb{N}$ for all $t \in T$. A *integer σ-run* of $\mathcal{N}$ is a sequence of states $\sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \sigma_2 \xrightarrow{a_2}_r \ldots$, where $\sigma_0 = \sigma \in \Sigma$ and $a_i \in T \cup \mathbb{N}$ for each $i \geq 0$. Note that all the states of an integer-run which starts at an integer-state are integer-states as well. Thus, it is easy to see that the following holds:

Lemma 1. *For a given time Petri net $\mathcal{N}$ the model $M_r(\mathcal{N})$ reduced to the integer-states and the transition relation between them is equal to $M_n(\mathcal{N})$.*

Given a number $x \in \mathbb{R}_+$, let $\lfloor x \rfloor$ denote the *floor* of x, i.e., the greatest $a \in \mathbb{N}$ such that $a \leq x$. Moreover, let $fire(\sigma)$ denote a set of transition that are ready to fire in the state $\sigma \in \Sigma$, i.e., $fire(\sigma) = \{t \in en(m) \mid clock(t) \in [Eft(t), Lft(t)]\}$. We define the integer-states to be *neighbour states* of real-valued ones as follows:

Definition 2 (Neighbour states). *Let $\sigma = (m, clock)$ be a state of a TPN $\mathcal{N}$. An integer-state $\sigma' = (m', clock')$ is a* neighbour state of σ *(denoted $\sigma' \sim_n \sigma$) iff*

- $m' = m$,
- $fire(\sigma) \subseteq fire(\sigma')$,
- *for each* $t \in en(m)$, $\lfloor clock(t) \rfloor \leq clock'(t) \leq \lfloor clock(t) \rfloor + 1$.

Intuitively, a neighbour state of σ is an integer-state of the same marking, and such that the values of its clocks, for all the enabled transitions, are "in a neighbourhood" of these of σ. However, these values can be such that they make more transitions ready to fire than the corresponding values in σ do.

Next, let $\pi := \sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \ldots$ be a σ^0-run in $M_r(\mathcal{N})$. We assign a time δ_i to each step $\sigma_i \xrightarrow{a_i}_r \sigma_{i+1}$ in the run, i.e., define $\delta_i = a_i$ if $a_i \in \mathbb{R}$, and $\delta_i = 0$ otherwise. Moreover, by $\Delta_G(\sigma_i, \pi)$, for $i \in \mathbb{N}$, we denote the value $\Sigma_{j=0}^{i-1}\delta_i$ (i.e., the time passed along π before reaching σ_i). We have the following theorem:

Theorem 1 (Thm 3.1 of [2]). *Let* $\pi := \sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \sigma_2 \xrightarrow{a_2}_r \ldots \xrightarrow{a_{k-1}}_r \sigma_k \xrightarrow{a_k}_r \ldots$ *be a dense* σ^0*-run of a TPN* $\mathcal{N}$*, and let* $\sigma = \sigma_k$ *be a state reachable along this run. There exists an integer* σ^0*-run* $\pi' := \sigma'_0 \xrightarrow{a'_0}_r \sigma'_1 \xrightarrow{a'_1}_r \sigma'_2 \ldots \xrightarrow{a'_{k-1}}_r \sigma'_k \xrightarrow{a'_k}_r \ldots$ *of* $\mathcal{N}$ *such that for each* $i = 0, \ldots, k$ *it holds*

1. $\sigma'_i \sim_n \sigma_i$
2. *if* $a_i \in T$ *then* $a'_i = a_i$.

Intuitively, the theorem states that if a state σ of $\mathcal{N}$ is reachable in k steps, then its neighbour integer-state σ' is reachable in k steps as well, by firing the same sequence of transitions and by time passings chosen in such a way that all the states visited to reach σ' are neighbour states of these visited to reach σ. So, we have:

Corollary 1. *If a state* σ *is reachable in* $M_r(\mathcal{N})$ *then there is an integer-state* σ' *which is reachable in* $M_n(\mathcal{N})$ *and such that* $\sigma' \sim_n \sigma$.

Concerning the construction of the run π', we have the following remark:

Remark 1. A run π' in Theorem 1 can be constructed in such a way that for each $i = 1, \ldots, k$ we have $\lfloor \Delta_G(\sigma_i, \pi) \rfloor = \Delta_G(\sigma'_i, \pi')$ (see [2]).

In fact, the paper [2] deals with time Petri nets with finite values of the Lft function, and in the runs considered the time- and action steps alternate. However, the proof of that theorem can be repeated without changes also in the case of non-alternating runs and TPNs with infinite Lfts.

Testing branching-time properties requires considering runs starting from arbitrary (reachable) states. Thus, we define the notion of a *neighbour run* of an arbitrary state of $M_r(\mathcal{N})$:

Definition 3 (Neighbour run). *Let* π *be a* σ*-run in* $M_r(\mathcal{N})$ *of the form* $\sigma = \sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \ldots$. *A run* π' $\sigma'_0 \xrightarrow{a'_0}_r \sigma'_1 \xrightarrow{a'_1}_r \ldots$ *is a* neighbour run *of* π *iff for each* $i \in \mathbb{N}$ *it holds*

- $\sigma'_i \sim_n \sigma_i$,

– *if* $a_i \in T$ *then* $a'_i = a_i$.

It is obvious from the above definition that each neighbour run of a σ-run starts at a neighbour state of σ and is an integer-run.

Given a run $\pi := \sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \ldots$, let $\pi_{[k]}$, for $k \in \mathbb{N}$, denote its prefix $\sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \ldots \xrightarrow{a_{k-1}}_r \sigma_k$. Moreover, given a σ_k-run $\rho := \sigma_k \xrightarrow{b_1}_r \ldots$, the run $\sigma_0 \xrightarrow{a_0}_r \ldots \xrightarrow{a_{k-1}}_r \sigma_k \xrightarrow{b_1}_r \ldots$ is denoted by $\pi_{[i]} \cdot \rho$. The next lemma shows how to construct some neighbour runs for the runs starting at a reachable state[2]:

Lemma 2 (Construction of some neighbour runs for reachable states). *Let $\sigma \in \Sigma$ be a reachable state of a TPN $\mathcal{N}$ such that $\sigma = \sigma_k$ for some σ^0-run $\pi := \sigma_0 \xrightarrow{a_0}_r \sigma_1 \xrightarrow{a_1}_r \ldots \xrightarrow{a_{k-1}}_r \sigma_k \xrightarrow{a_k}_r \ldots$, and let $\rho := \gamma_0 \xrightarrow{b_0}_r \gamma_1 \xrightarrow{b_1}_r \ldots$ be a σ-run of $\mathcal{N}$. The run $\rho' := \gamma'_0 \xrightarrow{b'_0}_r \gamma'_1 \xrightarrow{b'_1}_r \ldots$ such that for each $j \in \mathbb{N}$ it holds $b'_j = b_j$ if $b_j \in T$, and $\lfloor \Delta_G(\gamma_j, \pi_{[k]} \cdot \rho) \rfloor = \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho')$, where $\pi' := \sigma'_0 \xrightarrow{a'_0}_r \sigma'_1 \xrightarrow{a'_1}_r \ldots \xrightarrow{a'_{k-1}}_r \sigma'_k \xrightarrow{a'_k}_r \ldots$ denotes an integer σ^0-run with $\pi'_{[k]}$ constructed for $\pi_{[k]}$ according to Theorem 1 and Remark 1, is a neighbour run for ρ.*

Proof. The proof follows some constructions from the proofs of [2], and is by an induction on the length of the constructed prefix of the run ρ'.

From the fact that $\lfloor \Delta_G(\sigma_k, \pi_{[k]} \cdot \rho) \rfloor = \lfloor \Delta_G(\sigma_k, \pi) \rfloor = \Delta_G(\sigma'_k, \pi')$ and from the construction of π' (Remark 1) we have that $\gamma'_0 = \sigma'_k$ (which means that the run ρ' starts at the state of π' corresponding to σ_k) and therefore $\gamma'_0 \sim_n \gamma_0$.

Assume that $\rho'_{[j]}$ and $\rho_{[j]}$, of the states and labels of transitions satisfying the dependencies given in the lemma, are prefixes of neighbour runs. Consider the last state of $\rho'_{[j]}$, i.e. the state γ'_j of ρ' satisfying $\lfloor \Delta_G(\gamma_j, \pi_{[k]} \cdot \rho) \rfloor = \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho')$ and such that $\gamma'_j \sim_n \gamma_j$. Let for each $i \in \mathbb{N}$ $\gamma_i = (m_i, clock_i)$ and $\gamma'_i = (m'_i, clock'_i)$.

– if in ρ we have $b_j \in T$ then from the lemma it holds $b'_j = b_j$. Thus, the markings in γ_{j+1} and in γ'_{j+1} are equal. Let m_j denote the marking of γ_j. Consider a transition $t \in fire(\gamma_{j+1})$. If $t \notin newly\_en(m_j, b_j)$ then from $fire(\gamma_j) \subseteq fire(\gamma'_j)$ we have that $t \in fire(\gamma'_{j+1})$ (since firing b_j does not influence the value of the clock of t). If $t \in newly\_en(m_j, b_j)$ then in γ'_{j+1} the clock of t equals 0, which holds also in γ_{j+1}, and therefore either $t \in fire(\gamma_{i+1})$ and $t \in fire(\gamma'_{i+1})$, or $t \notin fire(\gamma_{i+1})$ and $t \notin fire(\gamma'_{i+1})$ Thus, $fire(\gamma_{j+1}) \subseteq fire(\gamma'_{j+1})$.
Consider $t \in en(m_j[b_j\rangle)$. Similarly as above, if $t \notin newly\_en(m_j, b_j)$ then the value of its clock in γ'_{j+1} is the same as in γ'_j, while if $t \in newly\_en(m_j, b_j)$

[2] The lemma is in fact a generalisation of the theorem of Popova (Thm 1): instead of reachability of a marking from the initial state, it considers reachability from an arbitrary reachable state, saying that a marking reachable from a reachable state σ is reachable from a neighbour state of σ on an integer-run. However, the current formalisation of the lemma is useful in the further part of the paper.

then its clock in γ'_{j+1} and in γ_{j+1} is 0. Thus from the fact that for γ_j, γ'_j we have $\lfloor clock_j(t) \rfloor \leq clock'_j(t) \leq \lfloor clock_j(t) \rfloor + 1$ we have also $\lfloor clock_{j+1}(t) \rfloor \leq clock'_{j+1}(t) \leq \lfloor clock_{j+1}(t) \rfloor + 1$, which implies $\gamma'_{j+1} \sim_n \gamma_{j+1}$.

– if $b_j \notin T$ then assume

$$b'_j = \lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho')$$

(which is an integer value due to $\Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho') \in \mathbb{N}$). We shall show that the time b'_j can pass at γ'_j leading to a state γ'_{j+1} such thath $\gamma'_{j+1} \sim_n \gamma_{j+1}$.

A) To show that b'_j can pass at γ'_j, notice that $b_j = \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) - \Delta_G(\gamma_j, \pi_{[k]} \cdot \rho)$. Moreover, we have that $clock_{j+1}(t) = clock_j(t) + b_j \leq Lft(t)$ for each $t \in en(m)$. Consider a transition $t \in en(m_j)$.
Let h be an index along $\pi_{[k]} \cdot \rho_{[j]}$ pointing to a state (denoted α) at which t became enabled most recently, and let h' be an index along $\pi'_{[k]} \cdot \rho'_{[j]}$ pointing to a state (denoted α') at which t became enabled most recently. From the equality of markings of the corresponding states in $\pi_{[k]} \cdot \rho$ and $\pi'_{[k]} \cdot \rho'$, and from the previous part of the proof (i.e., $b_j = b'_j$ for $b_j \in T$) we have that $h = h'$. It is easy to see that

$$clock_j(t) = \Delta_G(\gamma_j, \pi_{[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho)$$

and

$$clock'_j(t) = \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho') - \Delta_G(\alpha', \pi'_{[k]} \cdot \rho').$$

On the other hand,
$clock'_{j+1}(t) = clock'_j(t) + b'_j = \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho') - \Delta_G(\alpha', \pi'_{[k]} \cdot \rho') + b'_j =$
$\Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho') - \Delta_G(\alpha', \pi'_{[k]} \cdot \rho') + \lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \Delta_G(\gamma'_j, \pi'_{[k]} \cdot \rho') =$
$\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \Delta_G(\alpha', \pi'_{[k]} \cdot \rho') \overset{ind.hyp.\ and\ h=h'}{=}$
$\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor$

We have the following two cases:

- $clock_{j+1}(t) < Lft(t)$. This implies $\lfloor clock_{j+1}(t) \rfloor + 1 \leq Lft(t)$.
  From the property $\lfloor a \rfloor - \lfloor b \rfloor \leq \lfloor a - b \rfloor + 1$ (from [2]) we have $\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor \leq \lfloor \Delta_G(\gamma_{j+1,[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor + 1 = \lfloor clock_{j+1}(t) \rfloor + 1 \leq Lft(t)$ (from the inequality above).
- $clock_{j+1}(t) = Lft(t) \in \mathbb{N}$. In this case
  $\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor =$
  $\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho) + \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor =$
  $\lfloor clock_{j+1}(t) + \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor$.
  As we know that $clock_{j+1}(t) \in \mathbb{N}$, the above can be written as
  $clock_{j+1}(t) + \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor = clock_{j+1}(t) = Lft(t)$.

Summing up, we have that $clock'_j(t) + b'_j \leq Lft(t)$ for each $t \in en(m)$, and therefore the time b'_j can pass at γ'_j.

B) Next, we show that $\gamma'_{j+1} \sim_n \gamma_{j+1}$.
- Equality of their markings ($m'_{j+1} = m_{j+1}$) is obvious.

- To show that $fire(\gamma_{j+1}) \subseteq fire(\gamma'_{j+1})$ consider a transition $t \in en(m_{j+1})$. Assume that $t \in fire(\gamma_{j+1})$, which means that $clock_{j+1}(t) \in [Eft(t), Lft(t)]$. We need to show that $clock'_{j+1}(t) \geq Eft(t)$ (the condition $clock'_{j+1}(t) \leq Lft(t)$ is ensured by the previous part of the proof). Let α, α', h, h' be defined as in the part of the proof above. We have
  $Eft(t) \leq clock_{j+1}(t) = \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho)$. Since $Eft(t) \in \mathbb{N}$ then $Eft(t) \leq \lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor$. From the property $\lfloor a - b \rfloor \leq \lfloor a \rfloor - \lfloor b \rfloor$ ($a, b \in \mathbb{R}_+$ with $a \geq b$, [2]) we have $\lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) - \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor \leq \lfloor \Delta_G(\gamma_{j+1}, \pi_{[k]} \cdot \rho) \rfloor - \lfloor \Delta_G(\alpha, \pi_{[k]} \cdot \rho) \rfloor \overset{ind.hyp.\ and\ h=h'}{=} \Delta_G(\gamma'_{j+1}, \pi'_{[k]} \cdot \rho') - \Delta_G(\alpha', \pi'_{[k]} \cdot \rho') = clock'_{j+1}(t)$.
- finally, we need to show that for each $t \in en(m)$, $\lfloor clock_{j+1}(t) \rfloor \leq clock'_{j+1}(t) \leq \lfloor clock_{j+1}(t) \rfloor + 1$. The proof is a straightforward repetition of the appropriate elements of the part of the proof given above ($\lfloor clock_{j+1}(t) \rfloor \leq clock'_{j+1}(t)$ from the previous item in (B), $clock'_{j+1}(t) \leq \lfloor clock_{j+1}(t) \rfloor + 1$ from (A)).

We have the following corollary:

Corollary 2. *The σ^0-run $\pi'_{[k]} \cdot \rho'$ is a neighbour run for the σ^0-run $\pi_{[k]} \cdot \rho$.*

Having the above construction of neighbour runs, we shall show that given a state $\sigma \in \Sigma$ and a σ-run ρ which proves that a given $\mathrm{ECTL}^*_{-\mathrm{X}}$ formula φ holds at σ, one can find an integer-run ρ' which is a neighbour run for ρ and proves that φ holds at the initial state of ρ'.

Lemma 3 (Neighbour witness). *Let $M_r(\mathcal{N})$ be a dense model for a TPN $\mathcal{N}$, and φ be an arbitrary formula of $\mathrm{ECTL}^*_{-\mathrm{X}}$. Let ρ be a run starting at a reachable state σ of $M_r(\mathcal{N})$ and let ρ' be its neighbour run constructed as in Lemma 2, starting from an integer-state $\sigma' \sim_n \sigma$. The following conditions hold:*

1. *if $M_r(\mathcal{N}), \sigma \models \varphi$ then $M_r(\mathcal{N}), \sigma' \models \varphi$, where φ is a state formula, and*
2. *if $M_r(\mathcal{N}), \rho \models \varphi$ then $M_r(\mathcal{N}), \rho' \models \varphi$, where φ is a path formula.*

Proof. The proof is by an induction on the length of the formula: let φ be an $\mathrm{ECTL}^*_{-\mathrm{X}}$ formula; assume the lemma holds for all the proper subformulas of φ.

- Let $\varphi = p$ or $\varphi = \neg p$, where $p \in PV$. If $\sigma \models \varphi$ then from the equality of markings of the neighbour states we have also $\sigma' \models \varphi$.
- Let $\varphi = \mathrm{E}\psi$. If $\sigma \models \mathrm{E}\psi$ then $\rho \models \psi$ for some path ρ starting at σ. Consider the neighbour run ρ' for ρ, constructed as in Lemma 2, and denote $\rho'(0)$ (i.e., its starting state, satisfying $\rho'(0) \sim_n \sigma$ in an obvious way) by σ'. From the induction hypothesis we have $\rho' \models \psi$, which implies $\sigma' \models \mathrm{E}\psi$.
- Let $\varphi = \phi \wedge \psi$. Consider two cases:
  - φ is a state formula. If for a state $\sigma \in \Sigma$ it holds $\sigma \models \varphi$ then $\sigma \models \psi$ and $\sigma \models \phi$. From the induction hypothesis we have $\sigma' \models \phi$ and $\sigma' \models \psi$, which implies $\sigma' \models \varphi$;

  - φ is a path formula. If for a run ρ we have $\rho \models \varphi$ then $\rho \models \psi$ and $\rho \models \phi$. From the induction hypothesis for its neighbour run ρ' constructed as in Lemma 2 we have $\rho' \models \phi$ and $\rho' \models \psi$, and therefore $\rho' \models \psi \wedge \phi$.
- Let $\varphi = \phi \vee \psi$. Consider two cases:
  - φ is a state formula. If for a state $\sigma \in \Sigma$ we have $\sigma \models \varphi$ then $\sigma \models \psi$ or $\sigma \models \phi$. From the induction hypothesis it holds $\sigma' \models \phi$ or $\sigma' \models \psi$, which implies $\sigma' \models \varphi$;
  - φ is a path formula. If for a run ρ we have $\rho \models \varphi$ then $\rho \models \psi$ or $\rho \models \phi$. From the induction hypothesis for its neighbour run ρ' constructed as in Lemma 2 we have $\rho' \models \phi$ or $\rho' \models \psi$ and therefore $\rho' \models \psi \vee \phi$.
- Let $\varphi = \mathrm{G}\psi$. If $\rho \models \varphi$ then for each $j \geq 0$ it holds $\rho_j \models \psi$, where ρ_j is the j-th suffix of ρ. Since the starting state σ of ρ is reachable then the starting state of ρ_j, for each $j \geq 0$, is reachable as well. Moreover (as it is easy to see from the construction in Lemma 2), the neighbour run of ρ_j constructed as in Lemma 2 is equal to ρ'_j (i.e., when applying the construction from Lemma 2, the neighbour run for a j-th suffix of ρ is the j-th suffix of the neighbour run for ρ), and $\rho'_j(0) \sim_n \rho_j(0)$. Thus, the induction hypothesis can be applied to each suffix of ρ and the corresponding suffix of ρ'. Therefore, for each $j \geq 0$ it holds $\rho'_j \models \psi$, and therefore $\rho' \models \mathrm{G}\psi$.
- Let $\varphi = \phi \mathrm{U} \psi$. If $\rho \models \varphi$ then there is $j \geq 0$ such that $\rho_j \models \psi$ and for each $i = 0, \ldots, j-1$ we have $\rho_i \models \phi$. Analogously as in the proof for $\mathrm{G}\psi$ we can show for each $k = 0, \ldots, j$ we have that the starting state of ρ_k is reachable, and that the neighbour run for ρ_k is the k-th suffix of ρ', which implies that the induction hypothesis can be applied to each k-th suffix of ρ and the corresponding suffix of ρ'. Therefore, we have $\rho'_j \models \psi$ and $\rho'_i \models \phi$ for each $i = 0, \ldots, j-1$, which means that $\rho' \models \phi \mathrm{U} \psi$.

Thus, we can formulate the following corollary:

Corollary 3. *Let $M_r(\mathcal{N})$ and $M_n(\mathcal{N})$ be respectively a dense and a discrete model for a time Petri net $\mathcal{N}$, and let φ be an* $\mathrm{ECTL}^*_{-\mathrm{X}}$ *formula. If $M_r(\mathcal{N}) \models \varphi$ then $M_n(\mathcal{N}) \models \varphi$.*

The corollary follows from Lemma 1 and from the fact that a neighbour run constructed according to Lemma 2 for any σ^0-run is a σ^0-run as well.

On the other hand, due to the fact that all the states of $M_n(\mathcal{N})$ are also states of $M_r(\mathcal{N})$, and each run in $M_n(\mathcal{N})$ is a run on $M_r(\mathcal{N})$, we can formulate the following lemma:

Lemma 4. *Let $M_r(\mathcal{N})$ and $M_n(\mathcal{N})$ be respectively a dense and a discrete model for a time Petri net $\mathcal{N}$, and let φ be an* $\mathrm{ECTL}^*_{-\mathrm{X}}$ *formula. If $M_n(\mathcal{N}) \models \varphi$ then $M_r(\mathcal{N}) \models \varphi$.*

Finally, we have the following theorem:

Theorem 2. *Let $M_r(\mathcal{N})$ and $M_n(\mathcal{N})$ be respectively a dense and a discrete model for a time Petri net $\mathcal{N}$, and let φ be an* $\mathrm{ECTL}^*_{-\mathrm{X}}$ *formula. The following condition holds:*

$$M_r(\mathcal{N}) \models \varphi \text{ iff } M_n(\mathcal{N}) \models \varphi.$$

Proof. Follows from Lemmas 3 and 4 and Corollary 3 in a straightforward way.

Thus, for verification of $\mathrm{ECTL}^*_{-\mathrm{X}}$ formulas for time Petri nets the discrete model can be used instead of the dense one.

5 Experimental Results

In order to show that using discrete-timed models instead of the dense ones can be profitable we performed some tests, using as an example an implementation of SAT-based bounded model checking (BMC) for a subclass of TPNs (i.e. distributed time Petri nets) and a subclass of $\mathrm{ECTL}^*_{-\mathrm{X}}$ (i.e., $\mathrm{ECTL}_{-\mathrm{X}}$) [1]. The systems tested were the Generic Timed Pipeline Paradigm (GTPP) and the Fischer's mutual exclusion protocol (mutex), while the formulas were $\mathrm{EGEF}\mathit{ConsReceived}$, $\mathrm{EG}(\mathit{ProdReady} \vee \mathit{ConsReady})$ and $\mathrm{EF}\mathit{Node}_1\mathit{Send}$ for GTPP, and $\mathrm{EGEF}(\mathit{crit}_1 \vee \ldots \vee \mathit{crit}_n)$ for mutex (a description of the systems can be found in the appendix). The results are presented in Fig. 1–3 for GTPP, and in Fig. 4 for mutex. It can be seen that in all the cases we were able to verify systems containing more components (indicated in the column n) when discrete models were used, and the time ($\mathit{bmcT} + \mathit{satT}$) and the memory ($\max(\mathit{bmcM}, \mathit{satM})$) required are usually smaller for the discrete-time case (the columns with "$\mathbb{N}$:"). In some cases the differences are very substantial, but there are also examples in which the time and the memory used are similar for both the semantics. However, one can see that the noticeable differences occur in the cases in which the length of the witness for the formula (k) or the number of paths required to check this formula (LL) grow together with the growth of the system, making the verification more expensive.

n	k	LL	$\mathbb{R}$: bmcT+satT	$\mathbb{R}$: max(bmcM,satM)	$\mathbb{N}$: bmcT+satT	$\mathbb{N}$: max(bmcM,satM)
1	5	7	1.41	12.00	0.20	8.00
2	7	9	9.94	25.00	1.15	12.00
3	9	11	49.45	60.00	3.55	21.00
4	11	13	154.70	146.00	9.94	38.00
5	13	15	310.18	243.00	20.90	61.00
6	15	17	708.66	313.00	41.43	94.00
7	17	19	1934.63	818.00	76.81	145.00
8	19	21	4121.60	1071.00	131.98	215.00
9	21	23	6819.25	1640.00	237.21	314.00
10	23	25	20519.20	3455.00	361.03	377.00
11	25	27	-	-	562.15	552.00

Fig. 1. Comparison of the results for GTPP and the formula $\mathrm{EGEF}\mathit{ConsReceived}$

n	k	LL	$\mathbb{R}$: bmcT+satT	$\mathbb{R}$: max(bmcM,satM)	$\mathbb{N}$: bmcT+satT	$\mathbb{N}$: max(bmcM,satM)
1	5	1	0.41	7.00	0.17	7.00
2	7	1	4.14	12.00	0.76	8.00
3	9	1	32.27	29.00	2.20	9.00
4	11	1	63.28	52.00	8.57	12.00
5	13	1	200.14	151.00	21.14	17.00
6	15	1	488.59	165.00	43.18	24.00
7	17	1	870.21	342.00	105.18	38.00
8	19	1	1870.65	415.00	234.00	54.00
9	21	1	3745.33	658.00	763.84	139.00
10	23	1	7097.01	1364.00	1696.58	283.00
11	25	1	-	-	3013.98	306.00

Fig. 2. Comparison of the results: GTPP, the formula EG($ProdReady \vee ConsReady$)

n	k	LL	$\mathbb{R}$: bmcT+satT	$\mathbb{R}$: max(bmcM,satM)	$\mathbb{N}$: bmcT+satT	$\mathbb{N}$: max(bmcM,satM)
100	2	1	2.02	23.00	1.60	19.00
200	2	1	7.04	76.00	5.74	51.00
300	2	1	15.03	153.00	12.15	102.00
400	2	1	26.30	270.00	18.59	179.00
500	2	1	40.50	412.00	28.47	273.00
600	2	1	58.71	563.00	40.45	397.00
700	2	1	79.71	738.00	54.69	537.00
800	2	1	104.84	992.00	72.06	654.00
900	2	1	133.78	1173.00	90.48	854.00
1000	2	1	169.19	1528.00	114.86	1005.00
1100	2	1	211.16	1772.00	140.22	1168.00
1200	2	1	-	-	168.86	1506.00
1300	2	1	-	-	203.24	1604.00

Fig. 3. Comparison of the results: GTPP, the formula EF$Node1Send$

6 Conclusions and Further Work

We have shown that the result of Popova, stating that integer time steps are sufficient to test reachability of markings in time Petri nets, can be extended to testing $\mathrm{ECTL}^*_{-\mathrm{X}}$ properties. Our example experimental results prove that considering such a semantics can be profitable. Due to this, in our further work we are going to check whether discrete-time semantics can be used when testing other classes of properties of the dense-time Petri net systems.

References

1. A. Męski, W. Penczek, A. Półrola, B. Woźna-Szcześniak, and A. Zbrzezny. Bounded model checking approaches for verificaton of distributed time Petri nets. In *Proc.*

n	k	LL	$\mathbb{R}$: bmcT+satT	$\mathbb{R}$: max(bmcM,satM)	$\mathbb{N}$: bmcT+satT	$\mathbb{N}$: max(bmcM,satM)
2	4	5	1.04	11.00	0.74	10.00
3	4	5	1.77	13.00	1.10	12.00
4	4	5	1.83	15.00	1.63	14.00
5	4	5	3.18	17.00	2.41	16.00
10	4	5	9.15	40.00	7.20	31.00
20	4	5	26.14	90.00	18.76	86.00
30	4	5	74.70	177.00	58.56	161.00
40	4	5	258.34	330.00	108.34	320.00
50	4	5	265.06	419.00	170.93	358.00
60	4	5	710.11	732.00	442.42	713.00
70	4	5	701.81	1092.00	728.20	1073.00
80	4	5	919.34	1001.00	2288.34	1349.00
90	4	5	780.89	1161.00	934.72	1140.00
100	4	5	4566.16	3181.00	4230.64	4549.00
110	4	5	4260.76	3414.00	4956.38	3237.00
120	4	5	-	-	4217.44	3238.00
130	4	5	-	-	2155.04	2571.00
140	4	5	-	-	5087.76	3603.00

Fig. 4. Comparison of the results: mutex, the formula EGEF($crit_1 \vee \ldots \vee crit_n$)

of the Int. Workshop on Petri Nets and Software Engineering (PNSE'11), pages 72–91. University of Hamburg, 2011.
2. L. Popova. On time Petri nets. *Elektronische Informationsverarbeitung und Kybernetik*, 27(4):227–244, 1991.
3. L. Popova-Zeugmann and D. Schlatter. Analyzing paths in time Petri nets. *Fundamenta Informaticae*, 37(3):311–327, 1999.

7 Appendix

The first system we consider is the Generic Pipeline Paradigm Petri net model shown in Fig. 5. It consists of three parts: Producer producing data ($ProdReady$) or being inactive, Consumer receiving data ($ConsReady$) or being inactive, and a chain of n intermediate Nodes which can be ready for receiving data ($Node_iReady$), processing data ($Node_iProc$), or sending data ($Node_iSend$). The example can be scaled by adding more intermediate nodes. The parameters a, b, c, d, e, f are used to adjust the time properties of Producer, Consumer, and of the intermediate Nodes. The next system tested was the standard *Fischer's mutual exclusion protocol* (Mutex). The system consists of n time Petri nets, each one modelling a process, plus one additional net used to coordinate their access to the critical sections. A TPN modelling the system is shown in Fig. 6 for the case of $n = 2$.

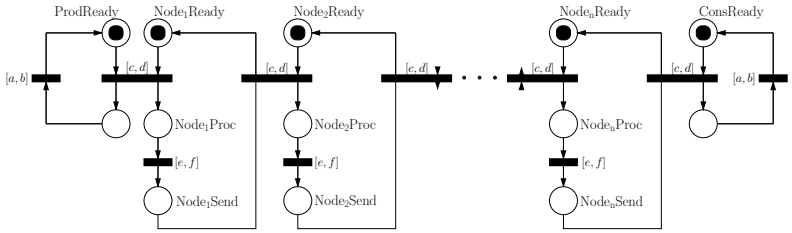

Fig. 5. A net for Generic Timed Pipeline Paradigm

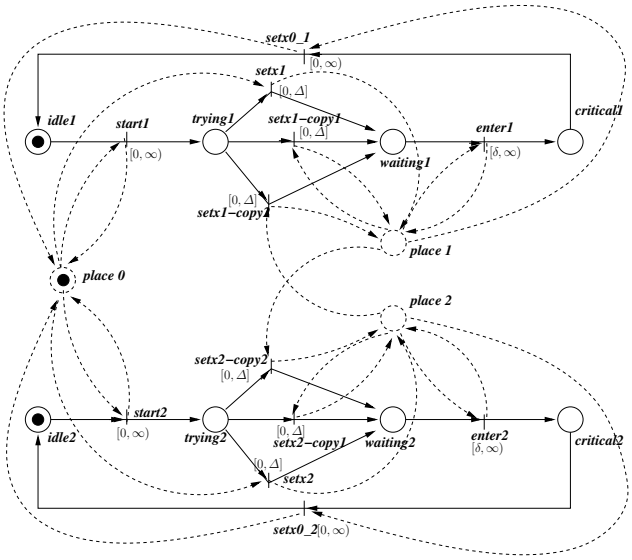

Fig. 6. A net for Fischer's mutual exclusion protocol for $n = 2$