Decidability of Zenoness, Token Liveness and Boundedness for Dense-Timed Petri Nets

Parosh Abdulla[1], Pritha Mahata[1], and Richard Mayr[2]

[1] Uppsala University, Sweden
[2] Albert-Ludwigs-University Freiburg, Germany
E-mail: {parosh,pritha}@it.uu.se,mayrri@informatik.uni-freiburg.de

Abstract. We consider *Timed Petri Nets (TPNs)* : extensions of Petri nets in which each token is equipped with a real-valued clock. We consider the following three verification problems for TPN.
(i) *Zenoness:* whether there is an infinite computation from a given marking which takes only a finite amount of time. We show decidability of zenoness for TPNs, thus solving an open problem from [dFERA00].
(ii) *Token liveness:* whether a token is *alive* in a marking, i.e., whether there is a computation from the marking which eventually consumes the token. We show decidability of the problem by reducing it to the *coverability problem*, which is decidable for TPNs.
(iii) *Boundedness:* whether the size of the reachable markings is bounded. We consider two versions of the problem; namely *semantic boundedness* where only live tokens are taken into consideration in the markings, and *syntactic boundedness* where also dead tokens are considered. We show undecidability of semantic boundedness, while we prove that syntactic boundedness is decidable through an extension of the Karp-Miller algorithm.

1 Introduction

Petri nets are one of the most widely used models for analysis and verification of concurrent systems. *Timed Petri nets (TPNs)* are extensions of Petri nets in the sense that each token has an "age" which is represented by a real valued clock (see [Bow96] for a survey). TPNs are computationally more powerful than timed automata [AD90], since they operate on a potentially unbounded number of clocks. This implies that TPNs can, among other things, model parameterized timed systems (systems consisting of an unbounded number of timed processes) [AN01]. Recently, several verification problems have been studied for TPNs (see e.g. [RGdFE99, dFERA00, AN01, AN02]). These problems are both extensions of classical problems previously studied for standard (untimed) Petri nets, and problems which are related to the timed behavior of TPNs. In this paper, we consider three verification problems for TPNs.

Zenoness A fundamental progress property for timed systems is that it should be possible for time to *diverge* [Tri99]. This requirement is justified by the fact that timed processes cannot be infinitely fast. Computations violating this property are called *zeno*. Given a TPN and a marking M, we check whether M is a *zeno-marking*, i.e., whether there is an infinite computation from M with a finite duration. The zenoness problem is solved in [Alu91] for timed automata using the region graph construction. Since region graphs only deal with a finite number of clocks, the algorithm of [Alu91] cannot

be extended to check zenoness for TPNs. In this paper, we solve the zenoness problem for TPNs. To do this, we consider a subclass of transfer nets [FS98] which we call *simultaneous-disjoint transfer net (SD-TN)*. This class is an extension of standard Petri nets, in which we also have *transfer* transitions which may move all tokens in one place to another with the restriction that (a) all such transfers take place simultaneously and (b) the sources and targets of all transfers are disjoint.
Given a TPN N, we perform the following three steps:
- Derive a corresponding SD-TN N'.
- Characterize the set of markings in N' from which there are infinite computations[3].
- Re-interpret the set computed above as a characterization of the set of zeno-markings in N.
In fact, the above procedure solves a more general problem than that of checking whether a given marking is zeno; namely it gives a characterization of the set of all zeno-markings.
The zenoness problem was left open in [dFERA00] both for dense TPNs (the model we consider in this paper) and for discrete TPNs (where behavior is interpreted over the discrete time domain). The construction given in this paper considers the dense case. The construction can easily be modified (in fact simplified) to deal with the discrete case.

Token Liveness Markings in TPNs may contain tokens which cannot be used by any future computations of the TPN. Such tokens do not affect the behavior of the TPN and are therefore called *dead tokens*. We give an algorithm to check, given a token and a marking, whether the token is dead (or alive). We do this by reducing the problem to the problem of *coverability* in TPNs. An algorithm to solve the coverability problem is given in [AN01].
Token liveness for dense TPNs was left open in [dFERA00].

Boundedness We consider the *boundedness problem* for TPNs: given a TPN and an initial marking, check whether the size of reachable markings is bounded. The decidability of this problem depends on whether we take dead tokens into consideration. In *syntactic boundedness* one considers dead tokens as part of the (size of the) marking, while in *semantic boundedness* we disregard dead tokens; that is we check whether we can reach markings with unboundedly many live tokens. Using techniques similar to [RGdFE99] it can be shown that semantic boundedness is undecidable. On the other hand we show decidability of syntactic boundedness. This is achieved through an extension of the Karp-Miller algorithm where each node represents a region (rather than a single marking). The underlying ordering on the nodes (regions) inside the Karp-Miller tree is a *well quasi-ordering* [Hig52]. This guarantees termination of the procedure.
Decidability of syntactic boundedness was shown for the simpler discrete case in [dFERA00], while the problem was left open for the dense case.

2 Definitions

Timed Petri Nets We consider *Timed Petri Nets* (*TPN*s) where each token is equipped with a real-valued clock representing the "age" of the token. The firing conditions of a

[3] In contrast to SD-TN, such a characterization is not computable for general transfer nets [May03].

transition include the usual ones for Petri nets. Additionally, each arc between a place and a transition is labeled with a subinterval of natural numbers. When firing a transition, tokens which are removed (added) from (to) places should have their ages lying in the intervals of corresponding arcs.

We use $\mathcal{N}, \mathcal{R}^{\geq 0}$ to denote the sets of natural numbers and nonnegative reals respectively. For a natural number k, we let $\mathcal{N}^k$ and $\mathcal{N}_\omega^k$ to denote the set of vectors of size k over $\mathcal{N}$ and $\mathcal{N} \cup \{\omega\}$ respectively.

We use a set *Intrv* of intervals. An open interval is written as $(w : z)$ where $w \in \mathcal{N}$ and $z \in \mathcal{N} \cup \{\infty\}$. Intervals can also be closed in one or both directions, e.g. $[w : z]$ is closed in both directions and $[w : z)$ is closed to the left and open to the right.

For a set A, we use A^* and $A^{\odot}$ to denote the set of finite words and finite multisets over A respectively. We view a multiset over A as a mapping from A to $\mathcal{N}$. Sometimes, we write multisets as lists, so $[2.4\ ,\ 5.1\ ,\ 5.1\ ,\ 2.4\ ,\ 2.4]$ represents a multiset b over $\mathcal{R}^{\geq 0}$ where $b(2.4) = 3$, $b(5.1) = 2$ and $b(x) = 0$ for $x \neq 2.4, 5.1$. For multisets b_1 and b_2 over $\mathcal{N}$, we say that $b_1 \leq b_2$ if $b_1(a) \leq b_2(a)$ for each $a \in A$. We define $b_1 + b_2$ to be the multiset b where $b(a) = b_1(a) + b_2(a)$, and (assuming $b_1 \leq b_2$) we define $b_2 - b_1$ to be the multiset b where $b(a) = b_2(a) - b_1(a)$, for each $a \in A$. We write $|b|$ for the number of elements in b. We use $\emptyset$ to denote the empty multiset and ϵ to denote the empty word.

Given a set A with an ordering $\preceq$ and a subset $B \subseteq A$, B is said to be *upward closed* if $a_1 \in B, a_2 \in A$ and $a_1 \preceq a_2$ implies $a_2 \in B$. Given a set $B \subseteq A$, we define the *upward closure* $B \uparrow$ to be the set $\{a \in A |\ \exists a' \in B :\ a' \preceq a\}$. A *downward closed* set B and the *downward closure* $B \downarrow$ are defined in a similar manner. We also use $a \uparrow$, $a \downarrow$, a instead of $\{a\} \uparrow$, $\{a\} \downarrow$, $\{a\}$ respectively.

Definition 1. *[AN01] A* Timed Petri Net (TPN) *is a tuple $N = (P, T, In, Out)$ where P is a finite set of places, T is a finite set of transitions and In, Out are partial functions from $T \times P$ to Intrv.*

If $In(t, p)$ (resp. $Out(t, p)$) is defined, we say that p is an *input (resp. output) place* of t.

We let max denote the maximum integer appearing on the arcs of TPN.

A *marking* M of N is a multiset over $P \times \mathcal{R}^{\geq 0}$. The marking M defines the numbers and ages of tokens in each place in the net. We identify a token in a marking M by the pair (p, x) representing its place and age in M. Then, $M((p, x))$ defines the number of tokens with age x in place p. Abusing notation again, we define, for each place p, a multiset $M(p)$ over $\mathcal{R}^{\geq 0}$, where $M(p)(x) = M((p, x))$.

For a set M of markings, we define $|\mathsf{M}| = max(|M| : M \in \mathsf{M})$.

For a marking M of the form $[(p_1, x_1)\ ,\ \ldots\ ,\ (p_n, x_n)]$ and $x \in \mathcal{R}^{\geq 0}$, we use M^{+x} to denote the marking $[(p_1, x_1 + x)\ ,\ \ldots\ ,\ (p_n, x_n + x)]$.

Transitions: We define two transition relations on the set of markings : timed transition and discrete transition. A *timed transition* increases the age of each token by the same real number. Formally, for $x \in \mathcal{R}^{\geq 0}$, $M_1 \longrightarrow_x M_2$ if $M_2 = M_1^{+x}$. We use $M_1 \longrightarrow_\delta M_2$ to denote that $M_1 \longrightarrow_x M_2$ for some $x \in \mathcal{R}^{\geq 0}$.

We define the set of *discrete transitions* $\longrightarrow_D$ as $\bigcup_{t \in T} \longrightarrow_t$, where $\longrightarrow_t$ represents the effect of *firing* the discrete transition t. More precisely, $M_1 \longrightarrow_t M_2$ if the set of input arcs $\{(p, \mathcal{I}) |\ In(t, p) = \mathcal{I}\}$ is of the form $\{(p_1, \mathcal{I}_1), \ldots, (p_k, \mathcal{I}_k)\}$, the set of output arcs $\{(p, \mathcal{I}) |\ Out(t, p) = \mathcal{I}\}$ is of the form $\{(q_1, \mathcal{J}_1), \ldots, (q_\ell, \mathcal{J}_\ell)\}$, and there are multisets

$b_1 = [(p_1, x_1), \ldots, (p_k, x_k)]$ and $b_2 = [(q_1, y_1), \ldots, (q_\ell, y_\ell)]$ over $P \times \mathcal{R}^{\geq 0}$ such that the following holds:
- $b_1 \leq M_1$
- $x_i \in \mathcal{I}_i$, for $i : 1 \leq i \leq k$.
- $y_i \in \mathcal{J}_i$, for $i : 1 \leq i \leq \ell$.
- $M_2 = (M_1 - b_1) + b_2$.

We say that t is *enabled* in M if there is a b_1 such that the first two conditions are satisfied. A transition t may be fired only if for each incoming arc, there is a token with the "right" age in the corresponding input place. These tokens will be removed when the transition is fired. The newly produced tokens may only have ages which belong to the relevant intervals.

We write $\longrightarrow = \longrightarrow_\delta \cup \longrightarrow_D$ to denote all transitions and use $\xrightarrow{*}$ to denote the reflexive transitive closure of $\longrightarrow$. It is easy to extend $\xrightarrow{*}$ for sets of markings. We define $Reach(M) := \{M' \mid M \xrightarrow{*} M'\}$ as the set of markings reachable from M.

A M-*computation* π from a marking M is a sequence $M_0, M_1, \ldots$ of markings s.t $M_0 = M$ and $M_i \longrightarrow M_{i+1}$ for $i \geq 0$.

For a computation π from M, the *delay* $\Delta(\pi)$ of the computation is defined as follows.

- $\Delta(M_0) = 0$.
- $\Delta(M_0, \ldots, M_{i+1}) = \Delta(M_0, \ldots, M_i)$ if $M_i \longrightarrow_D M_{i+1}$.
- $\Delta(M_0, \ldots, M_{i+1}) = \Delta(M_0, \ldots, M_i) + x$ if $M_i \longrightarrow_x M_{i+1}$.

Intuitively, the delay is the total amount of time passed in all timed transitions in the sequence. Notice that untimed Petri nets are a special case in our model where all intervals are of the form $[0 : \infty)$. Next, we recall a constraint system called *regions* defined for *Timed automata* [AD90].

Regions A *region* defines the integral parts of clock values up to max (the exact age of a token is irrelevant if it is greater than max), and also the ordering of the fractional parts. For TPNs, we need to use a variant which also defines the place in which each token (clock) resides. Following Godskesen [God94] we represent a region in the following manner.

Definition 2. *A region is a triple (b_0, w, b_{max}) where*

- $b_0 \in (P \times \{0, \ldots, max\})^\odot$. *$b_0$ is a multiset of pairs. A pair of the form (p, n) represents a token with age exactly n in place p.*
- $w \in \left((P \times \{0, \ldots, max-1\})^\odot\right)^*$. *$w$ is a word over the set $(P \times \{0, \ldots, max-1\})^\odot$, i.e., w is a word where each element in the word is a multiset over $P \times \{0, \ldots, max-1\}$. Pair (p, n) represents a token in place p with age x such that $x \in (n : n+1)$. Pairs in the same multiset represent tokens whose ages have equal fractional parts. The order of the multisets in w corresponds to the order of the fractional parts.*
- $b_{max} \in P^\odot$. *b_{max} is a multiset over P representing tokens with ages strictly greater than max. Since the actual ages of these tokens are irrelevant, the information about their ages is omitted in the representation.*

Formally, each region characterizes an infinite set of markings as follows. Assume a marking $M = [(p_1, x_1), \ldots, (p_n, x_n)]$ and a region $R = (b_0, b_1 b_2 \cdots b_m, b_{m+1})$. Let

b_j be of the form $[(q_{j1}, y_{j1}), \ldots, (q_{j\ell_j}, y_{j\ell_j})]$ for $j : 0 \leq j \leq m$ and b_{m+1} is of the form $[q_{m+1\ 1}, \ldots, q_{m+1\ l_{m+1}}]$. We say that M *satisfies* R, written $M \in R$, if there is a bijection h from the set $\{1, \ldots, n\}$ to the set of pairs
$\{(j,k) \mid (0 \leq j \leq m+1) \wedge (1 \leq k \leq \ell_j)\}$ such that the following conditions are satisfied:

- $p_i = q_{h(i)}$. Each token should have the same place as that required by the corresponding element in R.
- If $h(i) = (j,k)$ then $j = m+1$ iff $x_i > max$. Tokens older than max should correspond to elements in multiset b_{m+1}. The actual ages of these tokens are not relevant.
- If $x_i \leq max$ and $h(i) = (j,k)$ then $\lfloor x_i \rfloor = y_{j\ k}$. The integral part of the age of tokens should agree with the natural number specified by the corresponding elements in w.
- If $x_i \leq max$ and $h(i) = (j,k)$ then $frac(x_i) = 0$ iff $j = 0$. Tokens with zero fractional parts correspond to elements in multiset b_0.
- If $x_{i_1}, x_{i_2} < max$, $h(i_1) = (j_1, k_1)$ and $h(i_2) = (j_2, k_2)$ then $frac(x_{i_1}) \leq frac(x_{i_2})$ iff $j_1 \leq j_2$. Tokens with equal fractional parts correspond to elements in the same multiset (unless they belong to b_{m+1}). The ordering among multisets inside R reflects the ordering among fractional parts in clock values.

We let $[\![R]\!] = \{M \mid M \in R\}$.
Given a markings and a set $\mathbf{R}$ of regions, we say that $M \in \mathbf{R}$ if there is a region $R \in \mathbf{R}$ such that $M \in R$. We also let $[\![\mathbf{R}]\!] = \{M \mid \exists R \in \mathbf{R}.\ M \in R\}$.
The region construction defines an equivalence relation $\equiv$ on the set of markings such that $M_1 \equiv M_2$ if, for each region R, it is the case that $M_1 \in [\![R]\!]$ iff $M_2 \in [\![R]\!]$.
It is well-known [AD90] that $\equiv$ is a congruence on the set of markings. In other words, if $M_1 \longrightarrow M_2$ and $M_1 \equiv M_3$ then there is an M_4 such that $M_2 \equiv M_4$ and $M_3 \longrightarrow M_4$.
Notice that given a marking M, it is easy to compute the unique region $\mathbf{R}_M$ satisfied by M.
Ordering Next we define an ordering $\preceq$ on the set of markings such that $M_1 \preceq M_2$ if there is an M_2' with $M_1 \equiv M_2'$ and $M_2' \leq M_2$. In other words, $M_1 \preceq M_2$ if we can delete a number of tokens from M_2 and as a result obtain a new marking which is equivalent to M_1. We let $M_1 \prec M_2$ denote that $M_1 \preceq M_2$ and $M_1 \not\equiv M_2$. Notice that $\longrightarrow$ is *monotonic* with respect to the ordering $\preceq$, i.e, if $M_1 \longrightarrow M_2$ and $M_1 \preceq M_3$ then there is an M_4 such that $M_2 \preceq M_4$ and $M_3 \longrightarrow M_4$.
We identify a region R with the language $[\![R]\!]$ it represents (we write R instead of $[\![R]\!]$).
Next we show how to compute the language entailment of two regions. We lift $\preceq$ on the set of markings to the set of regions.

Definition 3. *Let $R = (b_0, b_1 \ldots b_m, b_{m+1})$ and $R' = (c_0, c_1 \ldots c_l, c_{l+1})$ be regions. Then, $R \preceq R'$ iff there is a strict monotone injection $g : \{0, \ldots, m+1\} \to \{0, \ldots, l+1\}$ with $g(0) = 0$ and $g(m+1) = l+1$ and $b_i \leq c_{g(i)}$ for each $i : 0 \leq i \leq m+1$.*

For regions R and R', if $R \preceq R'$ then for each $M \in [\![R]\!], M' \in [\![R']\!]$, we have $M \preceq M'$. We let $R \prec R'$ denote that $R \preceq R'$ and $R \neq R'$.

Lemma 4. *Union and intersection of (sets of) regions are computable as finite sets of regions.*

We define a function *Post* such that for a region R, *Post*(R) is a finite set of regions where a marking M' satisfies a region in *Post*(R) iff there is a marking $M \in [\![R]\!]$ such that $M \longrightarrow M'$. Thus *Post*(R) characterizes the set of markings we can reach from a marking satisfying R by a single transition.

Lemma 5. *The set Post*(R) *is effectively constructible.*

For a set $\mathbf{R}_1$ of regions, we define *Pre*$(\mathbf{R}_1)$ to be the set $\{R|\ \exists M_1 \in \mathbf{R}_1, M \in R.\ M \longrightarrow M_1\}$. We use min to denote a function which, given a set $\mathbf{R}_1$ of regions, returns the minimal elements of $\mathbf{R}_1$. We use $minpre([\![R]\!]\uparrow)$ to denote the set $\min(Pre(\{R\}\uparrow))$. We define Pre^* to be the reflexive, transitive closure of $minpre$.

Lemma 6. *Given a region R, the set $Pre^*([\![R]\!]\uparrow)$ is effectively constructible as a finite set of regions [AN01].*

3 Zenoness

A zeno-run of a timed Petri net is an infinite run that takes only finite time.

Definition 7. *Let $N = (P, T, In, Out)$ be a TPN. A marking M is called a* zeno-marking *iff there exists an infinite M-computation π and a finite number m s.t. $\Delta(\pi) < m$. Let $ZENO$ denote the set of all zeno-markings.*

The computability of the zenoness-problem for timed Petri nets (i.e., the problem if $M \in ZENO$ for a given marking M, or, more generally, constructing $ZENO$) was mentioned in [dFERA00] as an open problem for discrete-time Petri nets. Here we solve this problem even for the more general class of dense-time Petri nets. We show that, for any TPN, a decidable symbolic representation of the set $ZENO$ can be effectively computed. This implies the computability of $ZENO$ also for discrete-time Petri nets, since they can be encoded into dense-time ones. An alternative solution for discrete-time Petri nets is discussed in Remark 23.
The following brief outline explains the main steps of our proof.

1. We translate the original timed Petri net N into an untimed simultaneous-disjoint-transfer net N'. Simultaneous-disjoint-transfer nets are a subclass of transfer nets where all transfers happen at the same time and don't affect each other. The computations of N' represent, in a symbolic way, the computations of N that can be done in time $< 1 - \delta$ for some predefined $\delta > 0$.
2. Consider the set INF of markings of N' where infinite computations start. Since INF is upward-closed, it is characterized by its finitely many minimal elements; let INF_{min} be this set of minimal elements. While INF_{min} is not computable for general transfer nets [DJS99, May03], it is computable for simultaneous-disjoint-transfer nets, as shown in Lemma 21.
3. We re-interpret the set INF (resp. INF_{min}) of N' markings in the context of the timed Petri net N and construct from it the set $ZENO$.

Definition 8. A simultaneous-disjoint-transfer net (short SD-TN) N is described by a tuple $(P, T, Input, Output, Trans)$ where P is a set of places, T is a set of ordinary transitions, $Input, Output : T \rightarrow 2^P$ are functions that describe the input and output places of every transition, respectively. $Trans$ describes the simultaneous and disjoint transfer transition. We have $Trans = (I, O, S)$ where $I \subseteq P$, $O \subseteq P$, and $S \subseteq P \times P$. I and O describe the normal input and output places of the transfer transition and the pairs in S describe the source and target place of the transfer. The following restrictions on $Trans$ must be satisfied: If $(s,t), (s',t') \in S$ then s, s', t, t' are all different and $\{s,t\} \cap (I \cup O) = \emptyset$. Let $M : P \rightarrow \mathcal{N}$ be a marking of N. The firing of normal transitions $t \in T$ is defined just as for ordinary Petri nets. The transfer transition $Trans$ is enabled at M iff $\forall p \in I.\, M(p) \geq 1$. Firing $Trans$ yields the new marking M' where

$$\begin{array}{llll}
M'(p) = M(p) & \text{if } p \in I \cap O & M'(p) = M(p) - 1 & \text{if } p \in I - O \\
M'(p) = M(p) + 1 & \text{if } p \in O - I & M'(p) = 0 & \text{if } \exists p'.\, (p, p') \in S \\
M'(p) = M(p) + M(p') & \text{if } (p', p) \in S & M'(p) = M(p) & \text{otherwise}
\end{array}$$

(The restrictions above ensure that these cases are disjoint.)

Definition 9. For a given TPN $N = (P, T, In, Out)$ we now construct a SD-TN $N' = (P', T', Input, Output, Trans)$. The rough intuition is that N' simulates symbolically all computations of N which can happen in time $< 1 - \delta$ for some predefined $\delta > 0$.
Let $\max$ be the maximal finite constant that appears in the arcs of the TPN. We define a finite set of symbols $Sym := \{k \mid k \in \mathcal{N}, 0 \leq k \leq \max\} \cup \{k+ \mid k \in \mathcal{N}, 0 \leq k \leq \max\} \cup \{k- \mid k \in \mathcal{N}, 1 \leq k \leq \max\}$ and a total order on Sym by $k < k+ < (k+1)- < (k+1)$ for every k.
For every place $p \in P$ of N we have several corresponding places $p(s)$ for all $s \in Sym$ in P' of the SD-TN N'. Thus $P' = \{p(s) \mid p \in P, s \in Sym\}$. The intuition is as follows. A token on place $p(k)$ encodes a token of age exactly k on place p. A token on $p(k+)$ encodes a token on place p of an age a which satisfies $k < a \leq k + \delta$ for some a-priori defined $\delta > 0$. This means that the age of this token cannot reach $k + 1$ in any run taking time $< 1 - \delta$. A token on $p(k-)$ encodes a token on p whose age a satisfies $k - 1 + \delta < a < k$ and which may or may not reach age k during a run taking time $1 - \delta$. The SD-TN tokens $p(k), p(k+)$ and $p(k-)$ are called *symbolic encodings* of the corresponding TPN token (p, a).
In particular, the age of such a $p(k-)$ token could be chosen arbitrarily close to k, such that its age could reach (or even exceed) k in runs taking an arbitrarily small time. For example, consider a run taking time $\epsilon > 0$ and a transition with an outgoing arc labeled with the interval $[0 : 1)$. This transition could produce a token with age $1 - \epsilon/2$ which could reach age 1 or even $1 + \epsilon/2$ later.
We define a function $enc : Intrv \rightarrow 2^{Sym}$ as follows.

$$\begin{array}{ll}
enc([x : y]) := \{s \in Sym \mid x \leq s \leq y\} & enc((x : y]) := \{s \in Sym \mid x < s \leq y\} \\
enc([x : y)) := \{s \in Sym \mid x \leq s < y\} & enc((x : y)) := \{s \in Sym \mid x < s < y\}
\end{array}$$

For every transition $t \in T$ in the TPN N we have a set $T'(t)$ of new transitions in N'. The intuition is that the transitions in $T'(t)$ encode all possibilities of the age intervals of input and output tokens. For example, a discrete transition t with an input arc from place p labeled $[0 : 1]$ would yield 4 different transitions in $T'(t)$ with input arcs from places $p(0), p(0+), p(1-)$ and $p(1)$, respectively.
We denote the transitions with their sets of input and output places A and B by $t'(A, B)$, i.e., $Input(t'(A, B)) = A$ and $Output(t'(A, B)) = B$. For every $t \in T$ we define

$A(t) \subseteq 2^{P'}$ as follows.

$$A(t) := \{A \subseteq P' \mid In(t,p) \neq \bot \Leftrightarrow \exists_1 s \in enc(In(t,p)).\, p(s) \in A\}$$

For every $t \in T$ we define $B(t) \subseteq 2^{P'}$ as follows.

$$B(t) := \{B \subseteq P' \mid Out(t,p) \neq \bot \Leftrightarrow \exists_1 s \in enc(Out(t,p)).\, p(s) \in B\}$$

We use $\exists_1$ to mean there exists 'exactly one' in the above.
We define $T'(t) := \{t'(A,B) \mid A \in A(t), B \in B(t)\}$ and finally $T' := \bigcup_{t \in T} T'(t)$. Note that in transition $t'(A,B)$ of N', the set of input places A contains only one input place $p(s)$ for each non-empty interval $In(t,p)$ s.t. $s \in enc(In(t,p))$. A similar remark holds for the set of output places B.
So far, the transitions in T' only encode the discrete transitions of N. The transfer arc will be used to encode the passing of time. However, since we need to keep discreet transitions and time-passing separate, we must first modify the net to obtain alternating discrete phases and time-passing phases.
First we add two extra places p_{disc} and p_{time} to P' which act as control-states for the different phases. Then we modify all transitions $t \in T'$ by adding p_{disc} to $Input(t)$ and $Output(t)$. Thus normal transitions can fire iff p_{disc} is marked.
The transitions which encode the passing of time include both the transfer transition and several normal transitions. After an arbitrarily small amount of time < 1 passes, all tokens of age k have an age $> k$. This is encoded by the simultaneous-disjoint transfer arc, which moves all tokens from places $p(k)$ to places $p(k+)$. Formally, $Trans := (I, O, S)$ where $I := \{p_{disc}\}$, $O := \{p_{time}\}$, and $S := \{(p(k), p(k+)) \mid 0 \leq k \leq \max\}$. Note that $Trans$ starts the time-passing phase and switches the control-state from p_{disc} to p_{time}. Now we add a new set of transitions which encode what happens to tokens of age $k-$ when (a small amount of) time passes. Their age might either stay below k, reach k or exceed k. For every $k \in \{1, \ldots, \max\}$ we have a transition with input places p_{time} and $p(k-)$ and output places p_{time} and $p(k)$. Furthermore, for every $k \in \{1, \ldots, \max\}$ we have a transition with input places p_{time} and $p(k-)$ and output places p_{time} and $p(k+)$. Finally, we add an extra transition t_{switch} with input place p_{time} and output place p_{disc}, which switches the net back to normal discrete mode. Note that after a time-passing phase the only tokens on places $p(k)$ are those which came from $p(k-)$, because all tokens on $p(k)$ were first transferred to $p(k+)$ by the transfer transition.

Definition 10. *Let N be a TPN and N' the corresponding SD-TN, defined as in Def. 9. We say that a marking M' of N' is a* standard marking *if $M'(p_{disc}) = 1$ and $M'(p_{time}) = 0$. We denote by INF the set of all markings of N' from which infinite computations start. Since INF is upward-closed it can be characterized by its finitely many minimal elements. Let INF_{min} be the finite set of these minimal elements. Let INF' and INF'_{min} be the restriction to standard markings of INF and INF_{min}, respectively.*

The following definitions establish the connection between the markings of the timed Petri net N and the markings of the SD-TN N'.

Definition 11. *For every δ with $0 < \delta < 1$ we define a function $int_\delta : (P \times \mathcal{R}^{\geq 0})^{\odot} \to (P' \to \mathcal{N})$ that maps a marking M of N to its corresponding marking M' in N'.*

$M' := int_\delta(M)$ is defined as follows. For any k we have

$$\begin{array}{lll} M'(p(k)) & := M((p,k)) & M'(p_{disc}) := 1 \\ M'(p(k+)) & := \sum_{k<x\leq k+\delta} M((p,x)) & M'(p_{time}) := 0 \\ M'(p((k+1)-)) & := \sum_{k+\delta<x<k+1} M((p,x)) & \end{array}$$

Note that $M' = int_\delta(M)$ is a standard marking by Def. 10.

The intuition is as follows. In an infinite computation π starting at M with $\Delta(\pi) < 1-\delta$, no token (p,x) with $k < x \leq k+\delta$ can reach age $k+1$ by aging. This is reflected in N' by the fact that $p(k+)$ tokens stay $p(k+)$ tokens during the symbolic time-passing phase. On the other hand, tokens (p,x) with $k+\delta < x < k+1$ can reach an age $\geq k+1$ by aging. This is reflected in N' by the fact that $p((k+1)-)$ tokens can become $p(k+1)$ or $p((k+1)+)$ tokens during the symbolic time-passing phase.

Lemma 12. *Given a TPN N with marking M_0, the corresponding SD-TN N' from Def. 9 and $0 < \delta < 1$. If there exists an infinite M_0-computation π s.t. $\Delta(\pi) < 1-\delta$ then in N' there exists an infinite $int_\delta(M_0)$-computation π', i.e., $int_\delta(M_0) \in INF'$.*

The reverse implication of Lemma 12 does not generally hold. The fact that $int_\delta(M) \in INF'$ for some marking M of a TPN N does not imply that $M \in ZENO$. The infinite $int_\delta(M)$-run in N' depends on the fact that the $p(k-)$ tokens do (or don't) become $p(k)$ or $p(k+)$ tokens at the right step in the computation.
To establish a reverse correspondence between markings of N' and markings of N we need the following definitions.

Definition 13. *Let N' be a SD-TN with places $P' = \{p(s) \mid p \in P, s \in Sym\} \cup \{p_{disc}, p_{time}\}$ and a standard marking $M' : P' \to \mathcal{N}$. Let $M'^- := M'_{|\{p(k-) \mid k- \in Sym\}}$ and $M'^+ := M'_{|\{p(k+) \mid k+ \in Sym\}}$ be the sub-markings of M, restricted to $p(k-)$ and $p(k+)$ tokens, respectively. We define $perm(M'^-)$ as the set of all words $w = b_1 \ldots b_n \in \left((P \times \{0,\ldots,max-1\})^{\odot}\right)^*$ s.t. for all p and $k < max$ we have that $\left(\bigcup_{1\leq i\leq n} b_i\right)((p,k)) = M'^-(p((k+1)-))$. Similarly, let $perm(M'^+)$ be the set of all words $w = b_1 \ldots b_n \in \left((P \times \{0,\ldots,max-1\})^{\odot}\right)^*$ s.t. for all p and $k < max$ we have $\left(\bigcup_{1\leq i\leq n} b_i\right)((p,k)) = M'^+(p(k+))$.*

Intuitively, $perm(M'^-)$ describes all possible permutations of the fractional parts of tokens in a TPN marking M which are symbolically encoded as $p(k-)$ tokens in the corresponding SD-TN standard marking M'. Every symbolic marking M' of the SD-TN defines a set of TPN markings, depending on which permutation of the $p(k-)$ tokens and $p(k+)$ tokens is chosen.

Definition 14. *Let N' be a SD-TN. Every standard marking $M' : P' \to \mathcal{N}$ induces a set of regions $R(M',x,y)$ given by $R(M',x,y) = (b_0, xy, b_{max})$, where $b_0((p,k)) = M'(p(k))$ for all p and all $k \leq max$, $x \in perm(M'^+)$, $y \in perm(M'^-)$ and $b_{max}(p) = M'(p(max+))$ for all p.*

Lemma 15. *Let N be a TPN with corresponding SD-TN N' and $M' \in \mathit{INF}'$.*

$$\exists y \in perm(M'^{-}).\, \forall x \in perm(M'^{+}).\, [\![R(M', x, y)]\!]\!\uparrow \subseteq \mathit{ZENO}$$

Definition 16. *Let N be a TPN with corresponding SD-TN N'.*

$$Z := \bigcup_{M' \in \mathit{INF}'_{min}} \; \bigcup_{x \in perm(M'^{+})} \; \bigcap_{y \in perm(M'^{-})} \mathit{Pre}^{*}([\![R(M', x, y)]\!]\!\uparrow)$$

Now we show that the set ZENO is is effectively constructible.

Lemma 17. $\mathit{ZENO} = [\![Z]\!]\uparrow$.

Now we show that, for any SD-TN, the set INF'_{min} is effectively constructible. For this we use a result by Valk and Jantzen.

Theorem 18. *[VJ85] A finite basis of an upward-closed set $K \subseteq \mathcal{N}^k$ is effectively computable iff for any vector $u \in \mathcal{N}_\omega^k$ the predicate $u\!\downarrow \cap K \neq \emptyset$ is decidable.*

Lemma 19. *For any SD-TN N with initial marking M_0 the coverability graph can be effectively constructed.*

Note that Lemma 19 implies that place-boundedness is decidable for simultaneous-disjoint transfer nets, while it is undecidable for general transfer nets [DJS99, May03].

Lemma 20. *Given an SD-TN N with k places and an ω-marking $M_0 \in \mathcal{N}_\omega^k$, it is decidable if $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$.*

Lemma 21. *For any SD-TN N' the set INF'_{min} can be effectively constructed.*

Proof. Since INF is upward-closed, we can, by Lemma 20 and Theorem 18, construct a finite basis of INF, i.e., the set INF_{min}. We obtain INF'_{min} by the restriction of INF_{min} to standard markings. □

Theorem 22. *Let N be a TPN. The set ZENO is effectively constructible.*

Proof. We first construct the SD-TN N' corresponding to N, according to Def. 9. Then we consider the set Z from Def. 16. We have $\mathit{ZENO} = [\![Z]\!]\uparrow$ by Lemma 17. The set Z is effectively constructible by Lemma 21, Definition 16 and Lemmas 6 and 4. □

Remark 23. To compute *ZENO* for discrete-time Petri nets (left open in [dFERA00]), one can simply remove the transfer-transition from the net N' in Def. 9. This modified construction would yield ZENO for the discrete-time case, because (unlike in the dense-time case) every infinite zeno-run in a discrete-time net has an infinite suffix taking no time at all.

4 Token Liveness

First, we define the *liveness* of a token in a marking.
Let M be a marking in a TPN N. A token in M is called *syntactically k-dead* if its age is $\geq k$. It is trivial to decide whether a token is k-dead from a marking.
A token is called *semantically live* from a marking M, if we can fire a sequence of transitions starting from M which eventually consumes the token. Formally, given a token (p, x) and a marking M, we say that (p, x) can be *consumed* in M if there is a transition t satisfying the following properties: (a) t is enabled in M, (b) $In(t, p)$ is defined and $x \in In(t, p)$.

Definition 24. *A token (p, x) in M is semantically live if there is a finite M-computation $\pi = MM_1 \cdots M_r$ such that $(p, x + \Delta(\pi))$ can be consumed in M_r.*

For a marking M, we let $L(M)$ be the multiset of live tokens in M.

SEMANTIC LIVENESS OF TOKENS IN TPN

Instance: A timed Petri net N with marking M and a token $(p, x) \in M$.
Question: Is (p, x) live, i.e., $(p, x) \in L(M)$?
We show decidability of the semantic token liveness problem by reducing it to the *coverability problem* for TPNs.

COVERABILITY PROBLEM

Instance: A TPN N, a finite set of initial markings M_{init} of N, and an upward closed set $\mathsf{M}_{fin}\uparrow$ of markings of N, where M_{fin} is finite.
Question: $\mathsf{M}_{init} \xrightarrow{*} \mathsf{M}_{fin}\uparrow$?

Theorem 25. *The coverability problem is decidable for TPN [AN01].*

Suppose that we are given a TPN $N = (P, T, In, Out)$ with marking M and a token $(p, x) \in M$. We shall translate the question of whether $(p, x) \in L(M)$ into (several instances of) the coverability problem. To do that, we construct a new TPN N' by adding a new place p^* to the set P. The new place is not input or output of any transition. Either there is no transition in N which has p as its input place. Then it is trivial that $(p, x) \notin L(M)$. Otherwise, we consider all instances of the coverability problem defined on N' such that

- M_{init} contains a single marking $M - (p, x) + (p^*, x)$.
- M_{fin} is the set of markings of the form $[(p_1, x_1), \ldots, (p_n, x_n), (p^*, x')]$ such that there is a transition t and
  - the set of input places of t is given by $\{p, p_1, \ldots, p_n\}$.
  - $x' \in In(t, p)$ and $x_i \in In(t, p_i)$ for each $i : 1 \leq i \leq n$.

In the construction above, we replace a token (p, x) in the initial marking by a token (p^*, x); we also replace a token (p, x') in the final marking with $x' \in In(t, p)$, by a token (p^*, x'). The fact that the token in the question is not consumed in any predecessor of a marking in M_{fin}, is simulated by moving the token into the place p^* (in both the initial and final markings), since $p^* \notin P$ and not an input or output place in N'. Therefore, the token is live in M of N iff the answer to the coverability problem is 'yes'.
From Theorem 25, we get the following.

Theorem 26. *The token liveness problem is decidable.*

5 Boundedness

In this section we consider the boundedness problem for TPNs.

<u>SYNTACTIC BOUNDEDNESS OF TPN</u>

Instance: A timed Petri net N with initial marking M_0.
Question: Is $|Reach(M_0)|$ bounded ?
We give an algorithm similar to the Karp-Miller algorithm [KM69] for solving the syntactic boundedness problem for TPNs. The algorithm builds a tree, where each node of the tree is labeled with a region. We build the tree successively, starting from the root, which is labeled with R_{M_0}: the unique region satisfied by M_0 (it is easy to compute this region). At each step we pick a leaf with label R and perform one of the following operations:

1. If $post(R)$ is empty we declare the current node *unsuccessful* and close the node.
2. If there is another (already generated) node which is labeled with R then declare the current node *duplicate* and close the node.
3. If there is a predecessor of the current node labeled with $R' \prec R$ then declare $|Reach(M_0)|$ unbounded (the TPN is unbounded), and terminate the procedure.
4. Otherwise, declare the current node as an *interior* node, add a set of successors to it, each labeled with an element in *Post*(R). This step is possible due to Lemma 5.

If the condition of step 3 is never satisfied during the construction of the tree, declare $|Reach(M_0)|$ bounded (the TPN is bounded).
The proof of correctness of the above algorithm is similar to that of original Karp-Miller construction [KM69]. The termination of the algorithm is guaranteed due to the fact that the ordering $\preceq$ on the set of regions is a well-quasi-ordering (follows from [Hig52]).

Theorem 27. *Syntactic boundedness is decidable for TPNs.*

From Theorem 27, it follows that termination is also decidable for TPNs.
Since dead tokens cannot influence the behavior of a TPN, one would like to abstract from them.
Let N be a TPN with marking M. Then we define the live part of TPN markings as $Reach^l(M) := \{L(M') \mid M' \xrightarrow{*} M'\}$, i.e, $Reach^l(M)$ is the set of reachable markings without dead tokens.

<u>SEMANTIC BOUNDEDNESS OF TPN</u>

Instance: A timed Petri net N with initial marking M_0.
Question: Is $|Reach^l(M_0)|$ bounded ?
Using slightly modified constructions of [RGdFE99] or [AN02], we can easily derive the undecidability of semantic boundedness even for dense-timed Petri nets.

References

[AD90] R. Alur and D. Dill. Automata for modelling real-time systems. In *Proc. ICALP '90*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335, 1990.

[ADM04] P. A. Abdulla, J. Deneux, and P. Mahata. Multi-clock timed networks. 2004. To appear in LICS'04. Available from http://user.it.uu.se/˜pritha/Papers/tn2reach.ps.gz.

[AJ03] Parosh Aziz Abdulla and Bengt Jonsson. Model checking of systems with many identical timed processes. *Theoretical Computer Science*, 290(1):241–264, 2003.

[Alu91] R. Alur. *Techniques for Automatic Verification of Real-Time Systems*. PhD thesis, Dept. of Computer Sciences, Stanford University, 1991.

[AN01] P. A. Abdulla and A. Nylén. Timed Petri nets and BQOs. In *Proc. ICATPN'2001: 22nd Int. Conf. on application and theory of Petri nets*, volume 2075 of *Lecture Notes in Computer Science*, pages 53 –70, 2001.

[AN02] P. A. Abdulla and A. Nylén. Undecidability of ltl for timed petri nets. In *INFINITY 2002, 4th International Workshop on Verification of Infinite-State Systems*, 2002.

[Ber80] L. Berman. The complexity of logical theories. *TCS*, 11:71–77, 1980.

[BM99] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. of STACS'99*, volume 1563 of *LNCS*. Springer Verlag, 1999.

[Bow96] F. D. J. Bowden. Modelling time in Petri nets. In *Proc. Second Australian-Japan Workshop on Stochastic Models*, 1996.

[dFERA00] D. de Frutos Escrig, V. Valero Ruiz, and O. Marroquín Alonso. Decidability of properties of timed-arc Petri nets. In *ICATPN 2000*, number 1825 in Lecture Notes in Computer Science, pages 187–206, 2000.

[Dic13] L.E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *American Journal of Mathematics*, 35:413–422, 1913.

[DJS99] C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of Reset P/T Nets. In *Proc. of ICALP'99*, volume 1644 of *LNCS*. Springer Verlag, 1999.

[FS98] A. Finkel and P. Schnoebelen. Fundamental structures in well-structured infinite transition sytems. In *Proc. LATIN '98*, volume 1380 of *Lecture Notes in Computer Science*, pages 102–118, 1998.

[God94] J.C. Godskesen. *Timed Modal Specifications*. PhD thesis, Aalborg University, 1994.

[GS66] S. Ginsburg and E. Spanier. Semigroups, Presburger formulas, and languages. *Pacific J. of Mathematics*, 16:285–296, 1966.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.*, 2:326–336, 1952.

[KM69] R.M. Karp and R.E. Miller. Parallel program schemata. *Journal of Computer and Systems Sciences*, 3(2):147–195, May 1969.

[May03] R. Mayr. Undecidable problems in unreliable computations. *TCS*, 297(1-3):337–354, 2003.

[Par66] R. J. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

[RGdFE99] V. Valero Ruiz, F. Cuartero Gomez, and D. de Frutos Escrig. On non-decidability of reachability for timed-arc Petri nets. In *Proc. 8th International Workshop on Petri Nets and Performance Models*, pages 188–196, 1999.

[Tri99] S. Tripakis. Verifying progress in times systems. In *Proc. ARTS '99*, pages 299–314, 1999.

[VJ85] R. Valk and M. Jantzen. The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets. *Acta Informatica*, 21:643–674, 1985.

Appendix : Proofs of Theorems

Proof of Lemma 4

Union is trivially computable for regions.

We show the intersection of regions. Given two regions $R = (b_0, w_1, b_{m+1})$, $R' = (b'_0, w_2, b'_{n+1})$, there is a set of regions $R \cap R'$ such that $L(R \cap R') = L(R) \cap L(R')$.

First we consider the intersection $b = b_1 \cap b_2$ of multisets b_1, b_2 over a finite set A.

For each $a \in A$, $b(a) = min(b_1(a), b_2(a))$. If for all $a \in A$, $b(a) = 0$, then $b_1 \cap b_2 = \emptyset$.

For words w_1, w_2, there is a set of words $w_1 \cap w_2$ such that $L(w_1 \cap w_2) = L(w_1) \cap L(w_2)$.

In case w_1 and w_2 are singletons (i.e multisets), $w_1 \cap w_2$ is same as intersection of two multisets. We inductively define $w_1 \cap\ w_2$ as a set of words as follows. If one of them is empty, then the intersection is empty. Assume $w_1 = b_1 w'_1$ and $w_2 = b_2 w'_2$ are both non-empty.

Then $w_1 \cap w_2 = \{(b_1 \cap b_2)(w'_1 \cap w'_2)\ ,\ (w_1 \cap w'_2)\ ,\ (w'_1 \cap w_2)\}$.

Thus $R \cap R'$ is given by a set of regions $\mathbf{R}$ such that

$$\mathbf{R} = \Big\{ \big(c_0, w', c_{max}\big) \mid c_0 = b_0 \cap b'_0, w' \in w_1 \cap w_2, c_{max} = b_{m+1} \cap b'_{n+1} \Big\}$$

The intersection of sets of regions is defined in a straightforward manner.

Proof of Lemma 5

We define *Post* $=$ *Post*$_D$ $\cup$ *Post*$_\delta$ where *Post*$_D(R)$ corresponds to firing discrete transitions, and *Post*$_\delta(R)$ corresponds to letting time pass. *Post*$_D$ is computed as the union of *Post*$_t$ for all transitions t in the TPN where *Post*$_t$ characterizes the effect of running t once.

Let $R = (b_0, b_1 \dots b_m, b_{m+1})$. An addition (subtraction) corresponds to adding (removing) a token in a certain age interval. Let $\mathcal{I}$ be an interval of the form $[w : z)$ and let R be a region of the form $R = (b_0, b_1 \dots b_m, b_{m+1})$. We define the *addition* $R \oplus (p, \mathcal{I})$ as as a set of regions (the addition of other types of intervals can be defined in a similar manner).

We define $R \oplus (p, \mathcal{I})$ to be the union of the following four sets:

1. A set containing $(b_0 + (p, n), b_1 \dots b_m, b_{m+1})$, for each $n : w \leq n < z$. This corresponds to adding tokens with zero fractional parts.
2. A set containing regions $(b_0, b_1 \dots b'_i \dots b_m, b_{m+1})$, where $1 \leq i \leq m$, $b'_i = b_i + (p, n)$, and $w \leq n < z$, $n < max$. Elements added according to this case corresponds to adding a token with a fractional part equal to that of some other token.
3. A set containing regions $(b_0, b_1 \dots b_i\ \ (p, n)\ \ \dots b_m, b_{m+1})$, where n satisfies the same conditions as in 2. In this case, the fractional part differs from all other tokens.
4. A singleton set containing $(b_0, b_1 \dots b_m, b_{m+1} + p)$ if $z = \infty$.

Given R and $\mathcal{I}$ of the above forms, we define the subtraction $R \ominus (p, \mathcal{I})$ as the union of the following sets:

1. A singleton set containing $(b_0 - (p, n), b_1 \dots b_m, b_{m+1})$, for each $n : w \leq n < z$. This corresponds to subtracting tokens with zero fractional parts.
2. A set containing regions $(b_0, b_1 \dots b'_i \dots b_m, b_{m+1})$ where $b'_i = b_i - (p, n)$, where $1 \leq i \leq m$ and $w \leq n < z$, $n < max$. This corresponds to subtracting tokens with non-zero fractional parts.
3. A singleton set containing $(b_0, b_1 \dots b_m, b_{m+1} - p)$ in case $z = \infty$. This corresponds to removing tokens with age greater than max.
4. If all the above sets are empty, then $R \ominus (p, \mathcal{I})$ is undefined.

We extend $\oplus$ to sets $\mathbf{R}$ of regions so $\bigcup_{R \in \mathbf{R}} R \oplus (p, \mathcal{I})$. Subtraction can be extended in a similar manner. Notice that for a set A of pairs (arcs) of the form $(p, \mathcal{I})$, we have $R \oplus A = \bigcup_{(p,\mathcal{I}) \in A} R \oplus (p, \mathcal{I})$.

Let $A_{in}(t)$ be the set of input arcs given by $\{(p, \mathcal{I}) |\ \mathit{In}(t, p) = \mathcal{I}\}$ and the set of output arcs $A_{out}(t)$ be given by $\{(p, \mathcal{I}) |\ \mathit{Out}(t, p) = \mathcal{I}\}$.

We define

$$\mathit{Post}_t(R) = (R \ominus A_{in}(t)) \oplus A_{out}(t)$$

.

We compute the post-image of R wrt time as follows. First, we define a function $post_\delta$, which we use later to define Post_δ. For a multiset b over $P \times \{0, \ldots, max\}$, we use $young(b)$ to denote the multiset of the tokens $p(n)$ in b such that $n < max$. We use $old(b)$ to denote a multiset over P such that a place $p \in old(b)$ iff $p(\max) \in b - young(b)$.

Definition 28. *For a region $R = (b_0, b_1 \ldots b_m, b_{m+1})$, we define $post_\delta(R)$ to be a new region as follows:*

- *if $b_0 \neq \emptyset$, then $post_\delta(R) = (\emptyset, (young(b_0))b_1 \ldots b_m, (b_{m+1} + old(b_0)))$.*
- *Otherwise, $post_\delta(R) = (b'_m, b_1 \ldots b_{m-1}, b_{m+1})$. Here, b'_m contains a token $(p, n+1)$ for each token $(p, n) \in b_m$.*

Notice that $post_\delta(R)$ is undefined in case R is of the form $(\emptyset, , b)$ where b is a multiset over P,i.e., all tokens have ages greater than max. We define $post^*_\delta(R)$ to be the set of all regions which we can get by repeatedly applying the function $post_\delta$ to R until we get to a point where we generate the region $(\emptyset, , b)$. It is straightforward to verify that such a region will be eventually generated (by increasing the age of the tokens, all tokens will eventually become old). This gives us the following lemma.

Lemma 29. *$post^*_\delta(\{R\})$ is computable and $\mathit{Post}_\delta(R) = post^*_\delta(R)$.*

The definition of *Post*$_t$ and *Post*$_\delta$ proves Lemma 5.

Proof of Lemma 12

Proof. We show that for every infinite M-computation $\pi = M_0 \to M_1 \to M_2 \to \ldots$ with $\Delta(\pi) < 1 - \delta$ there is a corresponding infinite computation in N' of the form $int_\delta(M_0) \to^* int_{\delta_1}(M_1) \to^* int_{\delta_2}(M_2) \to^* \ldots$ with $1 > \delta_{i+1} \geq \delta_i$ for all i. Let π_i be the infinite suffix of π starting at M_i. The values of δ_i will be defined s.t. $\Delta(\pi_i) < 1 - \delta_i$. (The condition $\delta_{i+1} \geq \delta_i$ is required, because $\Delta(\pi_{i+1}) \leq \Delta(\pi_i)$.)

For every discrete transition step $M_i \to_t M_{i+1}$ there exists a transition step in N' of the form $int_{\delta_i}(M_i) \to int_{\delta_{i+1}}(M_{i+1})$, where $\delta_{i+1} = \delta_i$ by Def. 9 and Def. 11.

For every timed transition step $M_i \to_{\delta'} M_{i+1}$ (with $\delta' < 1 - \delta_i$) we have $\delta_{i+1} = \delta_i + \delta'$. By Def. 9 and Def. 11 there is a sequence of transitions in N' (the encoding of the time-passing phase) of the form $int_{\delta_i}(M_i) \to^* int_{\delta_i+\delta'}(M_{i+1})$. Note in particular that if some token (p, x) with $k + \delta_i < x < k + 1$ reaches an age equal to (or greater than) $k + 1$ in the transition from M_i to M_{i+1} then its symbolic encoding $p((k+1)-)$ can be transformed into a token $p(k+1)$ or $p((k+1)+)$ in the time-passing phase of N'. Furthermore, all tokens in M_i with fractional part 0 are transformed into tokens with a positive fractional part in M_{i+1}. In N' this is encoded by the fact that all $p(k)$ tokens become $p(k+)$ tokens in the time-passing phase. $\square$

Proof of Lemma 15

Proof. Since $M' \in \mathit{INF}'$ there is an infinite M'-computation $\pi' = M' \rightarrow M'_1 \rightarrow M'_2 \rightarrow \ldots$. This computation contains a (possibly infinite) number of time-passing phases (where the control-token shifts to the place p_{time}) $tpp_1, tpp_2, \ldots$. Now consider the *original* $p(k-)$ tokens in M' which become $p(k)$ tokens or $p(k+)$ tokens in the i-th time-passing phase tpp_i. Other tokens which were *newly created* during the computation π' are not considered. Let α_i be the multiset of $p(k-)$ tokens in M' which become $p(k+)$ tokens in tpp_i and β_i the multiset of $p(k-)$ tokens in M' which become $p(k)$ tokens in tpp_i. (Note that this does not happen by the simultaneous discrete transfer transition, but by normal transitions in the time-passing phase.) We have $\alpha_i, \beta_i \leq M'^{-}$, but not necessarily $\bigcup_{i \in \mathcal{N}} \alpha_i \cup \beta_i = M'^{-}$, because $p(k-)$ tokens can also be used by normal transitions in the discrete phase or never become $p(k)$ or $p(k+)$ tokens at all. Let $\gamma := M'^{-} - \bigcup_{i \in \mathcal{N}} \alpha_i \cup \beta_i$. Since M'^{-} is finite, there exists a smallest number m s.t. $\alpha_i \cup \beta_i = \emptyset$ for all $i > m$. It follows that there exists a minimal number n s.t. π' has an infinite suffix π'' starting at M'_n, s.t. in π'' no *original* $p(k-)$ token of M' becomes a $p(k)$ or $p(k+)$ token.

We define $y \in perm(M'^{-})$ by $y := \gamma\beta_m\alpha_m \ldots \beta_1\alpha_1$.

We need to prove that $\forall x \in perm(M'^{+}).\, [\![R(M', x, y)]\!]\!\uparrow \subseteq \mathit{ZENO}$. For this it suffices to show that $[\![R(M', x, y)]\!] \subseteq \mathit{ZENO}$, because ZENO is upward-closed. Now let $x \in perm(M'^{+})$ and $M \in [\![R(M', x, y)]\!]$. We need to show that $M \in \mathit{ZENO}$, i.e., that there exists an infinite time-bounded M-computation π.

Since $M \in [\![R(M', x, y)]\!]$ there exists a δ with $0 < \delta < 1$ and $int_\delta(M) = M'$. We construct an infinite M-computation π with $\Delta(\pi) < 1 - \delta$. The computation π has the form $M \rightarrow M_{j_1} \rightarrow M_{j_2} \rightarrow \ldots$ where the sequence $\{j_i\}_{i \in \mathcal{N}}$ is a subsequence of $1, 2, \ldots$ and $int_{\delta_{j_i}}(M_{j_i}) = M'_{j_i}$ and $\delta_{j_i} = \delta + \Delta(M \rightarrow \ldots \rightarrow M_{j_i})$.

For every discrete transition in π' (i.e., not in the time-passing phase) of the form $M'_i \rightarrow M'_{i+1}$ where $int_{\delta_i}(M_i) = M'_i$ there is a corresponding discrete transition in π of the form $M_i \rightarrow M_{i+1}$ where $\delta_{i+1} = \delta_i$ and $int_{\delta_{i+1}}(M_{i+1}) = M'_{i+1}$. This follows directly from Def. 8.

Now we consider the x-th time-passing phase for $1 \leq x \leq m$. For every sequence of transitions $M'_i \rightarrow^* M'_l$ in π' representing the x-th time-passing phase there is a corresponding single time-transition in π of the form $M_i \rightarrow_{\epsilon_x} M_l$, where $int_{\delta_i}(M_i) = M'_i$, $\delta_l = \delta_i + \epsilon_x$ and $int_{\delta_l}(M_l) = M'_l$. The delay ϵ_x is chosen as $\epsilon_x := 1 - b_x$ where b_x is the fractional part of the age of those tokens in M_i which are mapped to β_x by int_{δ_i}. This ensures that in this time transition the right tokens (of those originally present in M) reach (those mapped to β_x) or exceed (those mapped to α_x) the next higher integer age. For the other tokens of M_i, which were newly created during π we can arbitrarily choose the values of their fractional parts, i.e., for every combination of these values there is a possible run which implements it. Thus one can assume that these fractional parts are conveniently chosen such that they do (or don't) reach (or exceed) the next higher integer age, just as required by the condition $int_{\delta_l}(M_l) = M'_l$. Since $int_\delta(M) = M'$, only those tokens in M with a fractional part $> \delta$ were mapped to $p(k-)$ tokens in M' and only those tokens can reach (or exceed) age k in π'. Thus we can choose the ϵ_x in such a way that $\sum_{x=1}^{m} \epsilon_x < 1 - \delta$. Thus we get $\lambda := (1 - \delta) - \sum_{x=1}^{m} \epsilon_x > 0$. (The quantity λ will be used to determine the ϵ_x for $x > m$.)

Now we consider the x-th time-passing phase for $x > m$. These are the time-passing phases in the infinite suffix π'' mentioned above. For them, it works like the case above, except that the delays ϵ_x do no longer depend on the initial marking M, because $\alpha_x \cup \beta_x = \emptyset$ for $x > m$. As shown above, none of the original tokens of M are involved in these x-th time-passing phases for $x > m$. The only tokens involved in this (reaching or exceeding the next higher integer age in this phase) are tokens newly generated in π (which have an age greater than δ and are mapped to $p(k-)$). The fractional parts of their ages can be chosen conveniently s.t. they reach or exceed the next higher integer age exactly as required for the correspondence with the computation π'. In

particular, their ages can be chosen arbitrarily close to the next higher integer age s.t. the required delay ϵ_x can be made arbitrarily small. We choose $\epsilon_x := (\lambda/2) * 2^{-x}$ for $x > m$.
So we obtain $\Delta(\pi) = \sum_{x \in \mathcal{N}} \epsilon_x = \sum_{1 \le x \le m} \epsilon_x + \sum_{x > m} \epsilon_x \le \sum_{1 \le x \le m} \epsilon_x + \lambda/2 < \sum_{1 \le x \le m} \epsilon_x + \lambda = 1 - \delta$ and thus $\Delta(\pi) < 1 - \delta$ as required and therefore $M \in ZENO$. □

Proof of Lemma 17

Proof. First we show that $[\![Z]\!] \uparrow \subseteq ZENO$. It is enough to show that $[\![Z]\!] \subseteq ZENO$. Let $M \in [\![Z]\!]$. Then there is an $M' \in INF'_{min}$ and an $x \in perm(M'^{+})$ s.t. $M \in \bigcap_{y \in perm(M'^{-})} Pre^*([\![R(M', x, y)]\!]\uparrow)$. We choose $y \in perm(M'^{-})$ according to Lemma 15 and so obtain $M \in Pre^*([\![R(M', x, y)]\!]\uparrow)$ and $[\![R(M', x, y)]\!]\uparrow \subseteq ZENO$. Thus $M \in ZENO$, since $Pre^*(ZENO) = ZENO$[4].
Now we show that $ZENO \subseteq [\![Z]\!] \uparrow$.
It is enough to show that $min(ZENO) \subseteq [\![Z]\!]$. Let $M \in min(ZENO)$. By Def. 7 there exists an infinite M-computation π and a finite number m s.t. $\Delta(\pi) < m$. Thus there exists a marking M_1 s.t. $M \rightarrow^* M_1$ and an infinite M_1-computation π_1 with $\Delta(\pi_1) < 1/2$. Since M_1 contains finitely many tokens and π_1 is infinite, there exists an infinite suffix of π_1 s.t. none of the original tokens of M_1 is used in this suffix. Thus there exist markings M_2, M_3 and M_4 and a finite computation π_2 s.t.

- $M_1 \xrightarrow{\pi_2} M_2 = M_3 + M_4$
- All tokens in M_3 were created during π_2.
- There is an infinite M_3-computation π_3 with $\Delta(\pi_2 \pi_3) < 1/2$ (and thus $\Delta(\pi_3) < 1/2$).

Let $M_3' := int_{1/2}(M_3)$. Then we have $M_3' \in INF'$ by Lemma 12. There exists a marking $M_3'' \in INF'_{min}$ s.t. $M_3' \ge M_3''$. There exist permutations $x \in perm(M_3'^{+})$ and $y \in perm(M_3'^{-})$ s.t. $M_3 \in [\![R(M_3', x, y)]\!]$. Since $M_3' \ge M_3''$ there also exist permutations $x' \in perm(M_3''^{+})$ and $y' \in perm(M_3''^{-})$ s.t. $M_3 \in [\![R(M_3'', x', y')]\!]\uparrow \supseteq [\![R(M_3', x, y)]\!]$. ($x'$ and y' are the restrictions of x and y to $M_3''^{+}$ and $M_3''^{-}$, respectively.)
Now consider all those tokens in M_3 which are mapped to $p(k-)$ tokens in M_3', i.e. those with a fractional part of their age which is $\ge 1/2$. These tokens were all created during π_2 and none of them had an integer age during π_2, because $\Delta(\pi_2) < 1/2$. Thus, the fractional parts of their ages are totally independent and any permutation is possible, i.e., for any permutation there is a run which implements it. Therefore, for every $y \in perm(M_3'^{-})$ there is a marking M_3^y s.t.

- $M_1 \rightarrow^* M_3^y + M_4$
- $M_3^y \in [\![R(M_3', x, y)]\!]$.

Since $M_3' \ge M_3''$ we get that for every $y' \in perm(M_3''^{-})$ there is a marking $M_3^{y'}$ s.t

- $M_1 \rightarrow^* M_3^{y'} + M_4$
- $M_3^{y'} \in [\![R(M_3'', x', y')]\!]\uparrow$.

It follows that

$$M \in \bigcap_{y' \in perm(M_3''^{-})} Pre^*([\![R(M_3'', x', y')]\!]\uparrow)$$

with $M_3'' \in INF'_{min}$ and $x' \in perm(M_3''^{+})$, and thus $M \in Z$. □

[4] $Pre^*(ZENO)$ is the set of all markings from which one can reach a marking in *ZENO* by one or more transitions.

Proof of Lemma 19

Proof. We use ω-markings from $\mathcal{N}_\omega^k$ (where k is the number of places) and $x + \omega = \omega + x = \omega + \omega$ for any $x \in \mathcal{N}$ as usual. One proceeds from M_0 similarly as in the Karp-Miller construction [KM69], except for the transfer arc.

1. If one encounters the case $M_1 \xrightarrow{\pi} M_2$ with $M_2 \geq M_1$, $M_2 \neq M_1$ and the transfer arc was not used in π, then replace M_2 by $M_2 + \omega(M_2 - M_1)$. Obviously, π can be repeated arbitrarily often to yield an arbitrarily high numbers of tokens on the places where M_2 is strictly bigger than M_1.
2. Let M_1 and M_2 be two markings reached just after transfers, i.e., $\xrightarrow{trans} M_1 \xrightarrow{\pi} M_1' \xrightarrow{trans} M_2$ (where π may contain other transfers). If $M_2 \geq M_1$ and $M_2 \neq M_1$ then we replace M_2 by $M_2 + \omega(M_2 - M_1)$. The sequence of transitions $\xrightarrow{\pi}\xrightarrow{trans}$ can be repeated arbitrarily often to yield arbitrarily high numbers of tokens on the places where M_2 is strictly bigger than M_1. This is possible, because in SD-TN the set of places which are sources of transfers and the set of places which are targets of transfers are disjoint by Def. 8. Thus the transfers in $\xrightarrow{\pi}\xrightarrow{trans}$ do not negatively affect those places p where $M_2(p) > M_1(p)$. This point does not carry over to general transfer nets. In particular, all transfer-target places, once marked by ω in this construction, will stay ω in the future. Furthermore, all transfer source places are empty after the transfer, since all transfers are simultaneous.
3. If one reaches an ω-marking encountered before, then one creates a loop.

It is easy to show that the so-generated coverability graph is finite. Every infinite computation contains either finitely many or infinitely many transfers. In the first case the finiteness of the constrcution is ensured by Dickson's Lemma [Dic13] and the rules 1. and 3. above. In the second case the finiteness is ensured by Dickson's Lemma [Dic13] and the rules 2. and 3. above. $\square$

Proof of Lemma 20

Proof. We show that if $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$ then this condition will be detected by the following construction. Furthermore, we prove that the construction does not yield any false positives.
Let $C = (G, \rightarrow)$ with $G \subseteq \mathcal{N}_\omega^k$ be the coverability graph of (N, M_0), which is computable by Lemma 19. If $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$ then there exists a marking $M \in \mathcal{N}^k$ with $M \leq M_0$ and $M \in \mathit{INF}$. Thus there exists an infinite M-computation π. Now there are two cases. Either there are finitely many or infinitely many transfers in π.
In the first case, the simultaneous transfer transition Trans is used only finitely often in π. Thus π has an infinite suffix π', starting at some marking M', which uses only normal Petri net transitions. Since $M \rightarrow^* M'$, there must exist some ω-marking $M'' \in G$ s.t. $M' \leq M''$. Let N' be the ordinary Petri net obtained from N by removing the transfer transition Trans. So π' is an infinite M'-run of N'. Let $I_{N'} \subseteq \mathcal{N}^k$ be the (upward-closed) set of markings from which infinite runs of N' start. So we have $M''\!\downarrow \cap I_{N'} \neq \emptyset$. Generally, we check the condition (Cond1)

$$\exists M'' \in G.\ M''\!\downarrow \cap I_{N'} \neq \emptyset$$

This is a problem about ordinary Petri nets and has been shown to be decidable in [BM99]. If there is a witness computation of N for $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$ with only finitely many transfers then the condition (Cond1) is true as shown above. The reverse implication is trivial, because $I_{N'} \subseteq \mathit{INF}$.
In the second case, the simultaneous transfer transition Trans is used infinitely often in π. By Def. 8 of SD-TN, the source places and target places of transfers are disjoint and thus after a simultaneous transfer all source places are empty. We call a marking an 'after transfer marking'

(AT-marking) if it is reached just after firing $Trans$. We represent markings as vectors in $\mathcal{N}^k$ of the form (transfer source places, other places). So AT-markings have the form $(\mathbf{0}, \boldsymbol{x})$ with $\mathbf{0} \in \mathcal{N}^{k'}$ and $x \in \mathcal{N}^{k''}$ with $k = k' + k''$. The corresponding markings in the coverability graph C are called ω-AT-markings and have the form $(\mathbf{0}, \boldsymbol{x})$ with $x \in \mathcal{N}_\omega^{k''}$.

In this construction we use semilinear languages, Presburger arithmetic and Parikh's theorem. Semilinear sets are finite unions of linear subsets of $\mathbb{Z}^n$ (for arbitrary n). It is known that a set is semilinear iff it is definable in Presburger arithmetic [GS66]. Presburger arithmetic is the first-order theory of the integers with addition and order, also denoted as $(\mathbb{Z}, +, \leq)$. Presburger arithmetic is decidable; see, e.g., [Ber80]. Parikh's theorem [Par66] shows that the Parikh-image (the set of vectors, each representing the numbers of the different letters in a word) of regular (and even context-free) languages is effectively semilinear.

As a shorthand notation, we work with $\mathbb{Z}_\omega = \mathbb{Z} \cup \{\omega\}$ instead of the usual $\mathbb{Z}$. This is not a problem, since Presburger-arithmetic on $\mathbb{Z}_\omega$ can easily be reduced to Presburger-arithmetic on $\mathbb{Z}$ as follows. For every variable x one adds an extra variable x' which is used in such a way that the original state $x = k < \omega$ is represented by $(x, x') = (k, 0)$ and the original state $x = \omega$ is represented by $(x, x') = (0, 1)$. It is easy to encode the usual properties like $\omega + k = \omega - k = \omega + \omega = \omega$.

For any pair of ω-AT-markings $A, B \in G$ we can effectively construct a semilinear set $E(A, B) \subseteq \mathbb{Z}_\omega^k$ which represents all possible effects of computations of the form $\pi''\, Trans$ with $A \stackrel{\pi''}{\to} \stackrel{Trans}{\longrightarrow} B$ where π'' does not contain $Trans$. This is done as follows.

First, we compute the semilinear sets $E'(A, X) \subseteq \mathbb{Z}_\omega^k$ for all $X \in G$ s.t. $X \stackrel{Trans}{\longrightarrow} B$ in the coverability graph C and $A \to^* X$ without using $Trans$. The sets $E'(A, X)$ are semilinear and effectively constructible, because C is a finite graph whose arcs are labeled with constant vectors in $\mathbb{Z}_\omega^k$ and by Parikh's theorem [Par66] the Parikh-image of regular languages is effectively semilinear (i.e., one can consider A as the initial- and X as the final state of a finite automaton).

Then we effectively (by computability of Presburger-arithmetic and its equivalence with semilinear languages [GS66]) obtain $E(A, B)$ as the set of all vectors $(\mathbf{0}, x_1, \ldots, x_{k''}) \in \mathbb{Z}_\omega^k$ s.t. there exists an $X \in G$ with $X \stackrel{Trans}{\longrightarrow} B$ and a vector $(y_1, \ldots, y_{k'}, z_1, \ldots, z_{k''}) \in E'(A, X)$ and $x_j = z_j + y_{j'}$ whenever $(p_{j'}, p_j) \in S$ (the S from Def. 8, i.e., whenever there is a transfer from place $p_{j'}$ to place p_j) and $x_j = z_j$ otherwise. The set $E(A, B)$ is expressible in Presburger-arithmetic, since we used just addition, first order quantification and the semilinear sets $E'(A, X)$ in its construction. (Note that x_i, y_i, z_i can be ω.)

Now we construct a new finite graph $C' = (G', \to)$ as follows. $G' \subseteq G$ is the set of ω-AT-markings in G. For $A, B \in G'$ we have $A \to B$ in C' iff $A \stackrel{\pi''}{\to} \stackrel{Trans}{\longrightarrow} B$ in C where π'' does not contain $Trans$. The arc between A and B is labeled with (a symbolic Presburger-arithmetic representation of the semilinear set) $E(A, B)$.

We check the following condition (Cond2).

$$\exists n \in \mathcal{N}.\, A_0, \ldots, A_n \in G'.\, A_0 \to A_1 \to \ldots \to A_n = A_0 \exists \boldsymbol{v_i} \in E(A_i, A_{i+1}). \sum_{i=0}^{n-1} \boldsymbol{v_i} \geq \mathbf{0}$$

Note that the A_i above do not need to be disjoint.

Now we show the decidability of condition (Cond2). We transform the graph C', whose arcs $A \to B$ are labeled with semilinear sets $E(A, B)$ into a new equivalent graph C'' whose arcs are labeled with constant vectors. Since $E(A, B)$ is effectively semilinear, it can be represented as a finite union of linear sets of the form

$$E(A, B) = \bigcup_{i=1,\ldots,m} \{u_i + k_1 * w_i^1 + \ldots + k_{n_i} * w_i^{n_i} \mid k_1, \ldots, k_{n_i} \in \mathcal{N}\}$$

with computable $u_i, w_i^1, \ldots, w_i^{n_i} \in \mathbb{Z}_\omega^k$. For any pair of nodes A, B in C' we replace the arc $A \to B$ with the following arcs.

$$\begin{array}{ll} A \xrightarrow{u_i} A_i & \text{for new nodes } A_i \text{ and } 1 \leq i \leq m \\ A_i \xrightarrow{w_i^j} A_i & \text{for } 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i \\ A_i \xrightarrow{0} B & \end{array}$$

Let $C'' = (G'', \to)$ be the graph obtained in this way. We get immediately that the following condition (Cond3) holds for C'' iff (Cond2) holds for C'.

$$\exists n \in \mathcal{N}.\, A_0, \ldots, A_n \in G''.\, A_0 \xrightarrow{v_0} A_1 \xrightarrow{v_1} \ldots \xrightarrow{v_{n-1}} A_n = A_0.\, \sum_{i=0}^{n-1} \boldsymbol{v_i} \geq \mathbf{0}$$

(Note that the A_i above do not need to be disjoint.)

The condition (Cond3) is decidable, since C'' is a finite graph and by Parikh's theorem [Par66] the Parikh-image of regular languages is effectively semilinear. (Just interpret C'' as a finite automaton and try out any $A_0 \in G''$ as initial and final state.) Thus (Cond2) is decidable.

Finally, we show that (Cond2) is sufficient and necessary for the existence of an infinite M-run for some $M \leq M_0$.

If Cond2 is satisfied then there exist markings $M \leq M_0$ and $A \leq A_0$ s.t. $M \to^* A$ (by definition of C) and an infinite A-computation π^ω where $\pi = A_0 \to^* A_1 \to^* \ldots \to^* A_n$.

Now we show the other direction. Assume that there is some $M \in \mathcal{N}^k$ with $M \leq M_0$ and $M \in \mathit{INF}$ and some infinite M-computation π which uses Trans infinitely often. Thus is contains infinitely many AT-markings. Thus, by Dickson's Lemma [Dic13], there is a run $M \to^* (\mathbf{0}, \boldsymbol{x_1}) \xrightarrow{\pi'} (\mathbf{0}, \boldsymbol{x_2})$ with $\boldsymbol{x_2} \geq \boldsymbol{x_1}$. Thus the total effect $E(\pi')$ of the computation π' is nonnegative. It follows that there exists an ω-AT-marking $A_0 \in G$ with $A_0 \geq (\mathbf{0}, \boldsymbol{x_2})$ s.t. we have $A_0 \xrightarrow{\pi'} A_0$ in C where the effect of π' is nonnegative, i.e., $E(\pi') \geq \mathbf{0}$. The computation π' can be decomposed into $\pi' = \pi_1\pi_2 \ldots \pi_n$ with $A_i \xrightarrow{\pi_i} A_{i+1}$ for $1 \leq i \leq n-1$ and $A_n = A_0$ where $\{A_0, \ldots, A_n\}$ is the set of ω-AT markings visited in π'. (I.e., each π_i contains the transfer transition just once at the end.) It follows that $A_0 \to A_1 \ldots \to A_n = A_0$ is a cyclic path in C' and $\boldsymbol{v_i} := E(\pi_i) \in E(A_i, A_{i+1})$ and $\sum_{i=0}^{n-1} \boldsymbol{v_i} = E(\pi') \geq \mathbf{0}$. Therefore the condition (Cond2) is satisfied.

Altogether we obtain that $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$ iff (Cond1) or (Cond2) is satisfied. Since both conditions are decidable, we obtain decidability of $M_0\!\downarrow \cap\, \mathit{INF} \neq \emptyset$. $\square$