IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2018, accepted July 23, 2018, date of publication July 31, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2861467

An Algorithm of Recognizing Unbounded Petri Nets With Semilinear Reachability Sets and Constructing Their Reachability Trees

DAN YOU[1,2], SHOUGUANG WANG[1], (Senior Member, IEEE), AND WENHUI WU[1]
[1]School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
[2]Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy

Corresponding author: Shouguang Wang (wsg5000@hotmail.com)

This work was supported in part by the Zhejiang Natural Science Foundation under Grant LY15F030003 and in part by the National Natural Science Foundation of China under Grant 61472361.

ABSTRACT Reachability is one of the significant properties of Petri nets (PNs). For unbounded PNs, how to determine their reachability is an open issue. In this paper, we propose an algorithm that determines whether the reachability set of an unbounded PN is semilinear. Moreover, in the case that the unbounded PN has semilinear reachability set, the proposed algorithm can construct a new reachability tree (NRT) that exactly characterizes its reachability set. In addition, it can be decided based on NRT whether an unbounded PN suffers from deadlocks. The results are illustrated via examples.

INDEX TERMS Discrete-event systems, Petri nets, reachability trees, unbounded nets.

I. INTRODUCTION

Petri nets (PNs), thanks to their compact graphical representation and powerful mathematical foundation, are viewed as an appropriate modelling tool to analyze and control discrete-event systems (DESs) like computer networks, workflow systems, automated manufacturing systems, and urban traffic systems. The analysis on reachability sets is of fundamental importance for PNs since a variety of properties such as liveness, boundedness, coverability and reversibility all can be checked via analyzing reachability sets [27]. The reachability set can be represented by a tree named the reachability tree (RT) that is a powerful tool to check properties of PNs intuitively and thus its construction is very important. However, it is challenging to construct a reachability tree that exactly characterizes the reachability set for unbounded PNs. This is because their reachability sets are infinite. Over the past fifty years, many works [4], [8]–[11], [14], [17], [19]–[25], [30], [31], [33] study the finite representation of the reachability tree for PNs with infinite reachability sets.

The work [11] done by Karp and Miller firstly proposes a finite reachability tree (FRT) where a special symbol ω is introduced that denotes an infinite component of a marking due to some transition firing loops. It is proved that FRT can be utilized to decide such properties as boundedness and safeness [15], [16]. Unfortunately, the introduction of symbol ω in FRT leads to information loss, which as a result invalidates its determination on whether a specific marking is reachable, as well as the property of deadlock-freedom of an unbounded PN. For a subclass of unbounded PNs, Hiraishi [8] uses Periodic vectors instead of symbol ω to develop reduced representation of vector state spaces with infinite states. However, the method is not very efficient since most of the time is spent on checking the inclusion of regions [8]. Hence there is still much work [4], [9], [10], [14], [17], [19]–[25] utilizing symbol ω to find finite representation of reachability trees for PNs with infinite reachability sets.

An augmented reachability tree (ART) is developed by Jeng and Peng [9], [10], which extends the ability of FRT of analyzing qualitative properties of unbounded PNs, e.g., liveness. Its basic idea is computing the minimal marking for every node of the tree. However, the complexity of such a computation is NP-hard and the construction of ART essentially relies on enumeration technique. Besides, ART is unfortunately applicable to one-place-unbounded nets only, i.e., nets containing only one unbounded place.

To avoid information loss, Wang [19] proposes a modified reachability tree (MRT), where the expression $k\omega_n + q$ rather than ω is adopted to denote infinite components of a marking. The work in [21] develops a computer program that can generate an MRT of a net automatically. Later, Wang *et al.* [20] prove that an MRT is a finite tree and verify its capability on deciding such properties as the reachability, deadlock-freedom and liveness. Counterexamples in [4] and [17], however, indicate that the marking set represented by an MRT is not necessarily equal to the reachability set. Due to the existence of spurious reachable markings, MRT fails to correctly decide the deadlock and liveness of some unbounded nets. Thereafter, for one-place-unbounded nets, Wang *et al.* [25] develop an improved reachability tree (IRT) whose marking set is exactly the same as the reachability set.

In [24], ω-independent unbounded nets are proposed by Wang *et al.*, which are a subclass of unbounded nets more general than one-place-unbounded nets. For such unbounded PNs, Li [14] and Wang *et al.* [24] construct a new modified reachability tree (NMRT) that exactly represents the reachability set of a net and thus can correctly determine deadlocks and liveness [22]. Note that the modified next-state function is adopted during the construction of NMRT, by which spurious reachable markings in [20] are avoided.

To our best knowledge, almost all the reachability trees in the literature, which exactly characterize reachability sets of unbounded PNs, are only applicable to a limited subclass of PNs whose reachability sets are semilinear. Motivated by this fact, a method is proposed in our recent work [43] that can construct a reachability tree for any unbounded PN whose reachability set is semilinear. In this paper, we develop a new algorithm that is capable of recognizing whether an unbounded PN is a net with a semilinear reachability set. In addition, for an unbounded PN with a semilinear reachability set, the proposed algorithm constructs a *new reachability tree* (NRT) that exactly characterizes the infinite reachability set of the PN.

The remainder of this paper is organized as follows. Section II provides related notions used in this paper. Section III introduces modified expressions of ω-numbers and ω-markings. Section IV proposes an algorithm that determines whether an unbounded PN has the semilinear reachability set and constructs an NRT for the unbounded PN with the semilinear reachability set. Section V concludes this paper.

II. PRELIMINARIES

A. PETRI NETS

A *Petri net* (PN) is a four-tuple $N = (P, T, F, W)$ where P and T are the sets of *places* and *transitions*, respectively and they are finite, non-empty, and disjoint sets. $F \subseteq (P \times T) \cup (T \times P)$ is the set of *flow relation* that are graphically denoted by directed arcs connecting places to transitions. The function $W: (P \times T) \cup (T \times P) \rightarrow \mathbf{N}$ assigns each arc a weight. Given $x, y \in P \cup T$, $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$ otherwise. The *preset* of a node $x \in P \cup T$ is ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$ and the *post-set* of a node $x \in P \cup T$ is $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. The *incidence matrix* of N is a matrix $[N]: P \times T \rightarrow \mathbf{Z}$ such that $[N](p, t) = W(t, p) - W(p, t)$. $[N](p, \bullet)$ ($[N](\bullet, t)$) denotes the incidence vector with respect to a place p (transition t), i.e., a row (column) in $[N]$.

An *ordinary marking* μ of N is a mapping from P to $\mathbf{N}$. $t \in T$ is *enabled* at μ if $\forall p \in {}^\bullet t$, $\mu(p) \geq W(p, t)$. An enabled transition t at marking μ can fire. $\mu[t\rangle\mu'$ denotes that the firing of t at μ leads to a new marking μ', where $\mu'(p) = \mu(p) + [N](p, t)$, $\forall p \in P$. Note that T^* denotes the set of all finite sequences of transitions in T, including the empty sequence ε, where the operation "$*$" is called *Kleene-closure*. A transition sequence $\sigma = t_1 t_2 \ldots t_k \in T^*$ is feasible from a marking μ_1 if there exist $\mu_1, \mu_2, \ldots, \mu_{k+1}$ such that $\mu_i[t_i\rangle\mu_{i+1}$, $\forall i \in \mathbf{N}_k = \{1, 2, \ldots, k\}$. We use $\mu_1[\sigma\rangle\mu_{k+1}$ to denote that μ_{k+1} is reachable from μ_1 by firing σ. The set of all reachable markings of N from the initial marking μ_0 is denoted by $R(N, \mu_0)$. The *Parikh vector* of a transition sequence σ is $v_\sigma: T \rightarrow \mathbf{N}$, which maps t in T to the number of occurrences of t in σ. For instance, suppose that $T = \{t_1, t_2, t_3, t_4\}$. Then, the Parikh vector of the transition sequence $\sigma = t_1 2 t_3$ is $v_\sigma = [1, 0, 2, 0]^T$.

A PN (N, μ_0) is *bounded* if the token count of each place p does not exceed a finite number $B \in \mathbf{Z}^+$ for any marking μ reachable from μ_0, i.e., $\mu(p) \leq B$. Otherwise, the net is *unbounded*.

B. ω–NUMBERS

In this subsection, we review the related notations of ω–numbers defined in [4], [17], [19]–[21], [24], and [25] .

A subset of integers S is called an *ω-number* if $\exists k \in \mathbf{Z}^+$, $n, q \in \mathbf{Z}$ such that $S = \{ik + q | i \geq n\}$. S can be uniquely expressed as $S = \omega(k, n, q) \equiv k\omega_n + q \equiv \{ik + q | k \in \mathbf{Z}^+, n \in \mathbf{Z}, 0 \leq q < k, i \geq n\}$, where $\omega(k, n, q)$ or $k\omega_n + q$ is called a *canonical* ω-number with k as its *base*, n as the*least bound*, and q as the *remainder*.

A vector $x \in \mathbf{Z}_\omega^n$ is called an *ω-vector* if at least one of its components is an ω-number, where $\mathbf{Z}_\omega$ is the set of integers and ω-numbers. Clearly, an ω-vector can be viewed as a set of ordinary integer vectors. A marking μ is called an *ω-marking* if it can be represented by an ω-vector. An ω-marking can be viewed as a set of ordinary markings.

At an ω-marking μ, $t \in T$ is *enabled* if t is enabled at all ordinary markings of μ; t is *not enabled* at μ if t is not enabled at any ordinary marking of μ; t is *conditionally enabled* at μ if it is not enabled at some ordinary markings of μ but enabled at any other ordinary markings of μ. Note that if t is enabled at μ and $\mu' \geq \mu$, it holds that t is enabled at μ'.

C. SEMILINEAR SETS [5], [6], [12]

Let C and D be two subsets of $\mathbf{N}^n$, and $L(C, D)$ be the set of all $x \in \mathbf{N}^n$ in the form $x = c + k_1 \cdot d_1 + k_2 \cdot d_2 \ldots + k_m \cdot d_m$, where $c \in C$, $\{d_1, d_2, \ldots, d_m\} = D$, and $k_1, k_2, \ldots, k_m \in \mathbf{N}$. $L(C, D)$ is said to be a *linear set* if C consists of exactly one element and D is finite (possibly empty). A subset of $\mathbf{N}^n$ is said to be *semilinear* if it is a finite union of linear sets.

We can see that the set consisting of a single ordinary marking is a linear set and thus the reachability set of a bounded PN is definitely semilinear. However, for unbounded PNs, their reachability sets may not be semilinear.

III. MODIFIED ω-NUMBERS AND ω-MARKINGS

In this section, we introduce modified ω-numbers and ω-markings, which are firstly proposed in our previous work [43]. In this work, we still use them to construct reachability trees. We present the motivation of making such modifications in the following.

It is clear that infinite marking sets have to be represented by ω-markings when we construct reachability trees for unbounded nets. However, the previous work [4], [17], [19]–[21], [24], [25] does not explicitly define an ω-marking. Consider the following two infinite marking sets:

$$\begin{aligned} A &= \{(x, y)|x \in \{1, 2, 3, \ldots\}, \quad y \in \{1, 2, 3, \ldots\}\} \\ &= \{(1, 1), (1, 2), (1, 3), \ldots, (2, 1), (2, 2), (2, 3), \ldots, \\ &\quad (3, 1), (3, 2), (3, 3), \ldots\}; \text{ and} \\ B &= \{(x, x)|x \in \{1, 2, 3, \ldots\}\} \\ &= \{(1, 1), (2, 2), (3, 3), (4, 4), \ldots\}. \end{aligned}$$

According to the work in [4], [17], [19]–[21], [24], and [25], A and B are represented by a same ω-marking, i.e., $\mu = (\omega_1, \omega_1)$.

From the above example, we can see that, given an ω-marking, we cannot determine which marking set the ω-marking represents if no further information is provided. In order to solve this problem, we introduce superscripts for ω-numbers. Particularly, the expression of an ω-number $S = \omega(k, n, q) \equiv k\omega_n + q \equiv \{ik + q|i \geq n\}$ [4], [17], [19]–[21], [24], [25] is modified as $S = \omega^{(a)}(k, n, q) \equiv k\omega_n^{(a)} + q \equiv \{(i^{(a)} + n)k + q|i^{(a)} \in \mathbf{N}\}$, where $a \in \mathbf{Z}^+$. Based on the modified ω-numbers, the set A can be represented by $\mu_1 = (\omega_1^{(1)}, \omega_1^{(2)}) \equiv \{(i^{(1)} + 1, i^{(2)} + 1)|i^{(1)}, i^{(2)} \in \mathbf{N}\}$, while B represented by $\mu_2 = (\omega_1^{(1)}, \omega_1^{(1)}) \equiv \{(i^{(1)}+1, i^{(1)} + 1)|i^{(1)} \in \mathbf{N}\}$. Note that ω-numbers in a marking are associated with each other if they have the same superscript. In addition, an ω-number is generalized in this work into the following expression:

$$\begin{aligned} S &= \omega^{(a1)}(k_1, n_1, q_1) + \omega^{(a2)}(k_2, n_2, q_2) \\ &\quad + \ldots + \omega^{(am)}(k_m, n_m, q_m) \\ &\equiv (k_1\omega_{n1}^{(a1)} + q_1) + (k_2\omega_{n2}^{(a2)} + q_2) + \ldots + (k_m\omega_{nm}^{(am)} + q_m) \\ &\equiv \{((i^{(a1)} + n_1)k_1 + q_1) + ((i^{(a2)} + n_2)k_2 + q_2) \\ &\quad + \ldots + (i^{(am)} + n_m)k_m + q_m)|i^{(a1)}, i^{(a2)}, \ldots, i^{(am)} \in \mathbf{N}\}, \\ &\qquad \text{where } a_1, a_2, \ldots, a_m \in \mathbf{Z}^+. \end{aligned}$$

For example, the numbers listed below are all ω-numbers.

$$\begin{aligned} S_1 &= \omega^{(1)}(2, 0, 1) \equiv 2\omega_0^{(1)} + 1 \\ &\equiv \{2i^{(1)} + 1|i^{(1)} \in \mathbf{N}\} = \{1, 3, 5, 7, \ldots\}; \\ S_2 &= \omega^{(2)}(3, 1, 1) \equiv 3\omega_1^{(2)} + 1 \\ &\equiv \{3(i^{(2)} + 1) + 1|i^{(2)} \in \mathbf{N}\} = \{4, 7, 10, 13, \ldots\}; \\ S_3 &= \omega^{(1)}(2, 0, 1) + \omega^{(2)}(3, 1, 1) \equiv (2\omega_0^{(1)} + 1) + (3\omega_1^{(2)} + 1) \\ &\equiv \{(2i^{(1)} + 1) + (3(i^{(2)} + 1) + 1)|i^{(1)}, i^{(2)} \in \mathbf{N}\} \\ &= \{5, 7, 8, 9, 10, 11, \ldots\}; \\ S_4 &= \omega^{(2)}(3, 1, 0) + \omega^{(3)}(4, 2, 1) \equiv (3\omega_1^{(2)}) + (4\omega_2^{(3)} + 1) \\ &\equiv \{(3(i^{(2)} + 1)) + (4(i^{(3)} + 2) + 1)|i^{(2)}, i^{(3)} \in \mathbf{N}\} \\ &= \{12, 15, 16, 18, 19, 20, \ldots\}; \\ S_5 &= \omega^{(2)}(3, 0, 2) + \omega^{(3)}(4, 2, 2) \equiv (3\omega_0^{(2)} + 2) + (4\omega_2^{(3)} + 2) \\ &\equiv \{(3i^{(2)} + 2) + (4(i^{(3)} + 2) + 2)|i^{(2)}, i^{(3)} \in \mathbf{N}\} \\ &= \{12, 15, 16, 18, 19, 20, \ldots\}. \end{aligned}$$

Note that S_4 and S_5 differ in form but represent a same set. Actually, both of them can be transformed into a form: $S_4 = S_5 = 3\omega_0^{(2)} + 4\omega_0^{(3)} + 12$. Consequently, the least bounds of any ω-number, in this work, are all set as zero and they are thereby omitted for simplicity. The modified expression of ω-numbers is defined formally as follows.

Definition 1: A subset of integer S is called an *ω-number* if $\exists q \in \mathbf{Z}, k_1, k_2, \ldots k_m \in \mathbf{N}$ and $k_1 + k_2 + \ldots + k_m \neq 0$ such that

$$\begin{aligned} S &= \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q) \\ &\equiv k_1\omega^{(1)} + k_2\omega^{(2)} + \ldots + k_m\omega^{(m)} + q \\ &\equiv \{i^{(1)}k_1 + i^{(2)}k_2 + \ldots + i^{(m)}k_m + q|i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}. \end{aligned}$$

$\boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q)$ or $k_1\omega^{(1)} + k_2\omega^{(2)} + \ldots + k_m\omega^{(m)} + q$ is called a *canonical* ω-number, in which $\omega^{(j)}$ is called an *ω-element* with superscript j, k_j the *base* related to $\omega^{(j)}$, $j \in \{1, 2, \ldots, m\}$ and q the *starting-value*. Moreover, we call an ω-number is one with *z-dimension* if there are z non-zero bases in it.

According to Definition 1, some ω-number examples are listed below.

$$\begin{aligned} S_1 &= \boldsymbol{\omega}(2^{(1)}; 1) \equiv 2\omega^{(1)} + 1 \equiv \{2i^{(1)} + 1|i^{(1)} \in \mathbf{N}\} \\ &= \{1, 3, 5, 7, \ldots\}; \\ S_2 &= \boldsymbol{\omega}(0^{(1)}, 3^{(2)}; 4) \equiv 3\omega^{(2)} + 4 \equiv \{3i^{(2)} + 4|i^{(2)} \in \mathbf{N}\} \\ &= \{4, 7, 10, 13, \ldots\}; \\ S_3 &= \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 5) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 5 \\ &\equiv \{2i^{(1)} + 3i^{(2)} + 5|i^{(1)}, i^{(2)} \in \mathbf{N}\} \\ &= \{5, 7, 8, 9, 10, 11, \ldots\}; \\ S_4 &= \boldsymbol{\omega}(0^{(1)}, 3^{(2)}, 4^{(3)}; 12) \equiv 3\omega^{(2)} + 4\omega^{(3)} + 12 \\ &\equiv \{3i^{(2)} + 4i^{(3)} + 12|i^{(2)}, i^{(3)} \in \mathbf{N}\} \\ &= \{12, 15, 16, 18, 19, 20, \ldots\}. \end{aligned}$$

Clearly, S_1 and S_2 are ω-numbers with one-dimension, and S_3 and S_4 with two-dimension.

Definition 2: Given an ω-number $S = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q) \equiv k_1\omega^{(1)} + k_2\omega^{(2)} + \ldots + k_m\omega^{(m)} + q$, it is called a *simple ω-number* if its dimension is one, and otherwise a *compound ω-number*.

For instance, $\boldsymbol{\omega}(2^{(1)}; 1) \equiv 2\omega^{(1)} + 1$ and $\boldsymbol{\omega}(0^{(1)}, 3^{(2)}; 4) \equiv 3\omega^{(2)} + 4$ are both simple ω-numbers while $\boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 5) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 5$ and $\boldsymbol{\omega}(0^{(1)}, 3^{(2)}, 4^{(3)}; 12) \equiv 3\omega^{(2)} + 4\omega^{(3)} + 12$ are compound ω-numbers.

Remark 1: The simple ω-numbers are essentially the same as ω-numbers defined in [4], [17], [19]–[21], [24], and [25] when we consider the sets that they represent. As for compound ω-numbers, the sets represented by them can not necessarily be represented by an ω-number defined in [4], [17], [19]–[21], [24], and [25]. For example, $\boldsymbol{\omega}(0^{(1)}, 3^{(2)}, 4^{(3)}; 12) \equiv 3\omega^{(2)}+4\omega^{(3)}+12$ is a compound ω-number representing the set $\{12, 15, 16, 18, 19, 20, \ldots\}$. However, we cannot find an ω-number defined in [4], [17], [19]–[21], [24], and [25], which can represent such a set. In a word, the ω-number defined in this work is an extension to that in [4], [17], [19]–[21], [24], and [25] in terms of the set it represents.

Definition 3: Let $S_1 = \boldsymbol{\omega}(k_{11}^{(1)}, k_{12}^{(2)}, \ldots, k_{1m}^{(m)}; q_1)$ and $S_2 = \boldsymbol{\omega}(k_{21}^{(1)}, k_{22}^{(2)}, \ldots, k_{2m}^{(m)}; q_2)$ be two ω-numbers. We say S_1 and S_2 are *ω-numbers with the same form* if $k_{1i} = k_{2i}$, $\forall i \in \{1, 2, \ldots, m\}$.

Consider ω-numbers $S_1 = \boldsymbol{\omega}(2^{(1)}, 1^{(2)}; 1) \equiv 2\omega^{(1)}+\omega^{(2)}+1$, $S_2 = \boldsymbol{\omega}(2^{(1)}, 1^{(2)}; 4) \equiv 2\omega^{(1)}+\omega^{(2)}+4$, and $S_3 = \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 4) \equiv 2\omega^{(1)}+3\omega^{(2)}+4$. Clearly, S_1 and S_2 have the same form, while S_2 and S_3 do not.

Definition 4 (Addition of an ω-Number and an Integer): Let $S = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q)$ and $a \in \mathbf{Z}$. $S + a = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q + a)$.

For example, $\boldsymbol{\omega}(100^{(1)}; 1)+2 = \boldsymbol{\omega}(100^{(1)}; 3)$.

Definition 5 (Addition of Two ω-Numbers): Let $S_1 = \boldsymbol{\omega}(k_{11}^{(1)}, k_{12}^{(2)}, \ldots, k_{1m}^{(m)}; q_1)$ and $S_2 = \boldsymbol{\omega}(k_{21}^{(1)}, k_{22}^{(2)}, \ldots, k_{2m}^{(m)}; q_2)$ be two ω-numbers. $S_1 + S_2 = \boldsymbol{\omega}((k_{11} + k_{12})^{(1)}, (k_{12} + k_{22})^{(2)}, \ldots, (k_{1m} + k_{2m})^{(m)}; q_1 + q_2)$.

Consider two ω-numbers $S_1 = \boldsymbol{\omega}(2^{(1)}; 1) \equiv 2\omega^{(1)} + 1$ and $S_2 = \boldsymbol{\omega}(2^{(1)}; 2) \equiv 2\omega^{(1)} + 2$. We have $S_1 + S_2 = \boldsymbol{\omega}(4^{(1)}; 3) \equiv 4\omega^{(1)} + 3$. Consider another two ω-numbers $S_3 = \boldsymbol{\omega}(1^{(1)}; 1) \equiv \omega^{(1)} + 1$ and $S_4 = \boldsymbol{\omega}(0^{(1)}, 2^{(2)}; 0) \equiv 2\omega^{(2)}$. We have $S_3 + S_4 = \boldsymbol{\omega}(1^{(1)}, 2^{(2)}; 1) \equiv \omega^{(1)} + 2\omega^{(2)} + 1$.

Definition 6(Comparison of Two ω-Numbers): Let $S_1 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_1)$ and $S_2 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_2)$ be two ω-numbers with the same form. We say $S_1 \geq S_2 (S_1 > S_2)$ if $q_1 \geq q_2 (q_1 > q_2)$.

For instance, $S_1 = \boldsymbol{\omega}(1^{(1)}, 2^{(2)}; 2) \equiv \omega^{(1)} + 2\omega^{(2)} + 2$ and $S_2 = \boldsymbol{\omega}(1^{(1)}, 2^{(2)}; 3) \equiv \omega^{(1)} + 2\omega^{(2)} + 3$ are two ω-numbers with the same form. We have $S_2 > S_1$ since $3 > 2$. Note that the determination on which ω-number is bigger is defined on ω-numbers with the same form only. In other words, we cannot compare two ω-numbers with different forms. For example, $\boldsymbol{\omega}(4^{(1)}; 2)$ and $\boldsymbol{\omega}(2^{(1)}; 2)$ are not comparable.

Property 1: Let $S_1 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_1)$ and $S_2 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_2)$ be two ω-numbers with the same form. $S_1 \subseteq S_2$ if and only if q_1-$q_2 = c_1k_1+c_2k_2+\ldots+c_mk_m$, $c_1, c_2, \ldots, c_m \in \mathbf{N}$.

Proof (Sufficiency): It is clear that $S_1 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_1) \equiv \{i^{(1)}k_1 + i^{(2)}k_2 + \ldots + i^{(m)}k_m + q_1 | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$ and $S_2 = \boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q_2) \equiv \{i^{(1)}k_1 + i^{(2)}k_2 + \ldots + i^{(m)}k_m + q_2 | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$. Since $q_1 - q_2 = c_1k_1 + c_2k_2 + \ldots + c_mk_m$, we have $S_1 = \{(i^{(1)} + c_1)k_1 + (i^{(2)} + c_2)k_2 + \ldots + (i^{(m)} + c_m)k_m + q_2 | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\} = \{i^{(1)}k_1 + i^{(2)}k_2 + \ldots + i^{(m)}k_m + q_2 | i^{(1)} \geq c_1, i^{(2)} \geq c_2, \ldots, i^{(m)} \geq c_m\}$. Obviously, $S_1 \subseteq S_2$ holds.

(Necessity): Since $S_1 \subseteq S_2$ and $q_1 \in S_1$, we have $q_1 \in S_2 = \{i^{(1)}k_1+i^{(2)}k_2+\ldots+i^{(m)}k_m+q_2 | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$. Clearly, $\exists c_1, c_2, \ldots, c_m \in \mathbf{N}$ such that q_1-$q_2 = c_1k_1+c_2k_2+\ldots+c_mk_m$. ■

Property 1 provides a necessary and sufficient condition, under which there exists an inclusion relation between two ω-numbers with the same form. For instance, $S_1 = \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 2) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 2$, $S_2 = \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 3) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 3$, and $S_3 = \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 4) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 4$ are three ω-numbers with the same form. We have $S_3 \subseteq S_1$ since $\exists c_1 = 1$ and $c_2 = 0$, such that $4\text{-}2 = 2c_1 + 3c_2$, and we have $S_2 \not\subseteq S_1$ since $\forall c_1, c_2 \in \mathbf{N}$, $3\text{-}2 \neq 2c_1 + 3c_2$. ■

Definition 7: Let $S_1 = \boldsymbol{\omega}(k_{11}^{(1)}, k_{12}^{(2)}, \ldots, k_{1m}^{(m)}; q_1)$ and $S_2 = \boldsymbol{\omega}(k_{21}^{(1)}, k_{22}^{(2)}, \ldots, k_{2m}^{(m)}; q_2)$ be two ω-numbers. We say that S_1 is independent of S_2 if $k_{1i} \bullet k_{2i} = 0$, $\forall i \in \mathbf{N}_m$.

For example, $S_1 = \boldsymbol{\omega}(2^{(1)}, 0^{(2)}; 1) \equiv 2\omega^{(1)} + 1$ is independent of $S_2 = \boldsymbol{\omega}(0^{(1)}, 3^{(2)}; 4) \equiv 3\omega^{(2)} + 4$, while S_1 is not independent of $S_3 = \boldsymbol{\omega}(2^{(1)}, 3^{(2)}; 5) \equiv 2\omega^{(1)} + 3\omega^{(2)} + 5$.

Based on the modified definition of ω-numbers, the ω-vector (resp., ω-marking) exactly represents only one ordinary vector set (resp., ordinary marking set).

Let $\mu = (S_1, S_2, \ldots, S_n)$ be a vector, where $\forall x \in \mathbf{N}_n$, $S_x = \{i^{(1)}k_{x1}+i^{(2)}k_{x2}+\ldots+i^{(m)}k_{xm}+q_x | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$, in which $q_x \in \mathbf{Z}$, $k_{xy} \in \mathbf{N}$, $\forall y \in \mathbf{N}_m$. Note that S_x is an integer, i.e., $S_x = q_x$ if $k_{xy} = 0$, $\forall y \in \mathbf{N}_m$, and otherwise an ω-number. Then, the set represented by μ can be expressed by matrixes, i.e.,

$$\begin{aligned}
\mu &= (S_1, S_2, \ldots, S_n) \\
&\equiv \{I_{1\times m} \bullet K_{m\times n} + Q_{1\times n} | I_{1\times m} \in \mathbf{N}_m\}, \text{ where} \\
I_{1\times m} &= (i^{(1)}, i^{(2)}, \ldots, i^{(m)}), \\
K_{m\times n} &= \begin{bmatrix} k_{11} & k_{21} & \cdots & k_{n1} \\ k_{12} & k_{22} & \cdots & k_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ k_{1m} & k_{2m} & \cdots & k_{nm} \end{bmatrix}, \\
Q_{1\times n} &= (q_1, q_2, \ldots, q_n), \text{ and} \\
& \quad k_{xy} \in \mathbf{N}, q_x \in \mathbf{Z}, \forall x \in \mathbf{N}_n, \forall y \in \mathbf{N}_m.
\end{aligned}$$

Note that $\mu = (S_1, S_2, \ldots, S_n)$ is an ordinary vector if $K_{m\times n} = 0$, and otherwise an ω-vector defined before.

Remark 2: We should point out an ω-vector (resp., ω-marking) defined in this work is essentially a linear set.

Definition 8: Let $\mu = (S_1, S_2, \ldots, S_n)$ be an ω-vector. We say that μ is an independent ω-vector if $\forall i, j \in \{1, 2, \ldots, n\}$ and $i \neq j$, S_i and S_j are independent of each other, and otherwise μ is a dependent ω-vector.

For example, $\mu_1 = (\omega^{(1)}, \omega^{(2)})$ is an independent ω-vector since $\omega^{(1)}$ is independent of $\omega^{(2)}$, while $\mu_2 = (\omega^{(1)}, \omega^{(1)} + \omega^{(2)})$ is a dependent one since $\omega^{(1)}$ is not independent of $\omega^{(1)} + \omega^{(2)}$.

Property 2: Let $\mu = (S_1, S_2, \ldots, S_n)$ be an ω-vector. We have $\mu = \Delta$ if and only if μ is an independent ω-vector, where $\Delta = \{(a_1, a_2, \ldots, a_n) | a_g \in S_g$ (or $a_g = S_g$ if S_g is an integer), $\forall g \in \{1, 2, \ldots, n\}\}$.

Proof (Sufficiency): The proof is trivial.

(Necessity): By contradiction, suppose that μ is not an independent ω-vector. Hence, $\exists x, y \in \{1, 2, \ldots, n\}$ and

$x \neq y$ such that S_x is not independent of S_y. Let $S_x = \{i^{(1)}k_{x1}+i^{(2)}k_{x2}+\ldots+i^{(m)}k_{xm}+q_x | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$ and $S_y = \{i^{(1)}k_{y1}+i^{(2)}k_{y2}+\ldots+i^{(m)}k_{ym}+q_y | i^{(1)}, i^{(2)}, \ldots, i^{(m)} \in \mathbf{N}\}$. According to Definition 7, $\exists j \in \{1, 2, \ldots, m\}$ such that $k_{xj} \bullet k_{yj} \neq 0$. Now, let $i^{(j)} = c \geq 1$ and $\forall h \neq j, i^{(h)} = 0$. In this case, $a_x = ck_{xj} + q_x$ and $a_y = ck_{yj} + q_y$. This means when $a_x = ck_{xj} + q_x \in S_x$, a_y cannot be equal to any number in S_y except $ck_{yj} + q_y$. Hence, $\mu \neq \Delta$, which obviously contradicts the fact $\mu = \Delta$. Therefore, μ is an independent ω-vector. ■

Here, Δ is a set consisting of ordinary vectors, which results from an arbitrary combination of n components that are either an element in a set represented by an ω-number or an integer.

Definition 9: Let $\mu_1 = (S_{11}, S_{12}, \ldots, S_{1n})$ and $\mu_2 = (S_{21}, S_{22}, \ldots, S_{2n})$ be two ω-vectors. We say μ_1 and μ_2 are *ω-vectors with the same form* if S_{1x} and S_{2x} are ω-numbers with the same form or both integers, $\forall x \in \mathbf{N}_n$.

For example, $\mu_1 = (\omega^{(1)} + 3\omega^{(2)} + 7, \omega^{(2)} + 8, 3)$ and $\mu_2 = (\omega^{(1)} + 3\omega^{(2)} + 4, \omega^{(2)} + 4, 1)$ are two ω-vectors with the same form, while $\mu_3 = (\omega^{(1)}, 1)$ and $\mu_4 = (\omega^{(1)}, \omega^{(2)})$ are not.

Definition 10 (Comparison of Two ω-Vectors): Let $\mu_1 = (S_{11}, S_{12}, \ldots, S_{1n})$ and $\mu_2 = (S_{21}, S_{22}, \ldots, S_{2n})$ be two ω-vectors with the same form. We say $\mu_2 \geq \mu_1$, if $S_{2i} \geq S_{1i}$, $\forall i \in \{1, 2, \ldots, n\}$. Note that $\mu_2 > \mu_1$ is defined as $\mu_2 \geq \mu_1$ but $\mu_2 \neq \mu_1$.

For example, $\mu_1 = (\omega^{(1)}3\omega^{(2)} + 7, \omega^{(2)} + 8, 1)$ and $\mu_2 = (\omega^{(1)} + 3\omega^{(2)} + 4, \omega^{(2)} + 4, 1)$ are two ω-vectors with the same form. We have $\mu_1 > \mu_2$, since $\omega^{(1)} + 3\omega^{(2)} + 7 > \omega^{(1)} + 3\omega^{(2)} + 4$, $\omega^{(2)} + 8 > \omega^{(2)} + 4$, and 1=1. Note that we do not compare $\mu_3 = (\omega^{(1)}, 1)$ and $\mu_4 = (\omega^{(1)}, \omega^{(2)})$ in this work since they are two ω-vectors with different forms.

As mentioned before, an ω-vector is actually an infinite set of ordinary vectors. Hence, it is necessary to explore the condition under which there is an inclusion relation between two ω-vectors.

First, consider two ω-vectors $\mu_1 = (\omega^{(1)} + 2\omega^{(2)}, \omega^{(1)})$ and $\mu_2 = (3\omega^{(1)} + 3\omega^{(2)}, \omega^{(1)})$. Since $\omega^{(1)} + 2\omega^{(2)} = \{0, 1, 2, 3, \ldots\} = \mathbf{N}$ and $3\omega^{(1)} + 3\omega^{(2)} = \{0, 3, 6, 9, \ldots\}$, we have $3\omega^{(1)} + 3\omega^{(2)} \subseteq \omega^{(1)} + 2\omega^{(2)}$. Besides, it is obvious that $\omega^{(1)} \subseteq \omega^{(1)}$. Surprisingly, however, there is no inclusion relation between μ_1 and μ_2. In particular, we can see that μ_1 contains ordinary vector (2, 0), which is not contained in μ_2 and μ_2 contains (3, 0), which is not contained in μ_1.

From the above example, we know that, given two ω-vectors μ_1 and μ_2, even if each component of μ_1 is contained in the corresponding component of μ_2, it does not necessarily hold that μ_1 is contained in μ_2. Fortunately, such a conclusion can hold for two independent ω-vectors.

Property 3: Let $\mu_1 = (S_{11}, S_{12}, \ldots, S_{1n})$ and $\mu_2 = (S_{21}, S_{22}, \ldots, S_{2n})$ be two independent ω-vectors. $\mu_1 \subseteq \mu_2$ if and only if $S_{1i} \subseteq S_{2i}$ or $S_{1i} = S_{2i}$ or $S_{1i} \in S_{2i}$, $\forall i \in \{1, 2, \ldots, n\}$.

Proof: Straightforward from Property 2. ■

For example, $\mu_1 = (2\omega^{(1)}, 2, 1)$ and $\mu_2 = (\omega^{(1)}, 2\omega^{(2)}, 1)$ are both independent ω-vectors. We have $\mu_1 \subseteq \mu_2$ since $2\omega^{(1)} \subseteq \omega^{(1)}$, $2 \in 2\omega^{(2)}$, and 1=1.

Property 4: Let $\mu_1 = (S_{11}, S_{12}, \ldots, S_{1x}, \ldots, S_{1n})$ and $\mu_2 = (S_{21}, S_{22}, \ldots, S_{2x}, \ldots, S_{2n})$ be two ω-vectors with the same form. $\mu_1 \subseteq \mu_2$ if and only if there exists $C_{1\times m} = (c_1, c_2, \ldots, c_m) \in \mathbf{N}^m$ such that $\forall x \in \{1, 2, \ldots, n\}$,

1) $q_x - q'_x = C_{1\times m} \bullet (k_{x1}, k_{x2}, \ldots, k_{xm})^T$ if $S_{1x} = \boldsymbol{\omega}(k_{x1}^{(1)}, k_{x2}^{(2)}, \ldots, k_{xm}^{(m)}; q_x)$ and $S_{2x} = \boldsymbol{\omega}(k_{x1}^{(1)}, k_{x2}^{(2)}, \ldots, k_{xm}^{(m)}; q'_x)$ are ω-numbers with the same form, and

2) $S_{1x} = S_{2x}$ if S_{1x} and S_{2x} are both integers.

Proof: Clearly, the sets represented by μ_1 and $\mu_2$are accordingly as follows.

$$\begin{aligned}\mu_1 &= (S_{11}, S_{12}, \ldots, S_{1x}, \ldots, S_{1n})\\ &\equiv \{I_{1\times m} \bullet K_{m\times n} + Q_{1\times n} | I_{1\times m} \in \mathbf{N}^m\}, \text{ and}\\ \mu_2 &= (S_{21}, S_{22}, \ldots, S_{2x}, \ldots, S_{2n})\\ &\equiv \{I_{1\times m} \bullet K_{m\times n} + Q'_{1\times n} | I_{1\times m} \in \mathbf{N}^m\},\end{aligned}$$

where

$$\begin{aligned}I_{1\times m} &= (i^{(1)}, i^{(2)}, \ldots, i^{(m)}),\\ K_{m\times n} &= \begin{bmatrix} k_{11} & k_{21} & \cdots & k_{n1}\\ k_{12} & k_{22} & \cdots & k_{n2}\\ \vdots & \vdots & \ddots & \vdots\\ k_{1m} & k_{2m} & \cdots & k_{nm}\end{bmatrix} \neq 0,\\ Q_{1\times n} &= (q_1, q_2, \ldots, q_n),\\ Q'_{1\times n} &= (q'_1, q'_2, \ldots, q'_n), \text{ and}\\ &\quad k_{xy} \in \mathbf{N}, q_x, q'_x \in \mathbf{Z},\\ &\quad \forall x \in \{1, 2, \ldots, n\}, \quad \forall y \in \{1, 2, \ldots, m\}.\end{aligned}$$

(Sufficiency): We can know that there exists $C_{1\times m} = (c_1, c_2, \ldots, c_m) \in \mathbf{N}^m$, such that $\forall x \in \{1, 2, \ldots, n\}$, $q_x - q'_x = C_{1\times m} \bullet (k_{x1}, k_{x2}, \ldots, k_{xm})^T$. Hence, we have

$$\begin{aligned}\mu_1 &= (S_{11}, S_{12}, \ldots, S_{1x}, \ldots, S_{1n})\\ &\equiv \{I_{1\times m} \bullet K_{m\times n} + Q_{1\times n} | I_{1\times m} \in \mathbf{N}^m\}\\ &= \{(I_{1\times m} + C_{1\times m}) \bullet K_{m\times n} + Q'_{1\times n} | I_{1\times m} \in \mathbf{N}^m\}.\end{aligned}$$

Obviously, $\mu_1 \subseteq \mu_2$ holds.

(Necessity): It is clear that $Q_{1\times n} \in \mu_1$. Since $\mu_1 \subseteq \mu_2$, we have $Q_{1\times n} \in \mu_2$. Hence, $\exists I_{1\times m} \in \mathbf{N}^m$, such that $Q_{1\times n} = I_{1\times m}\bullet K_{m\times n} + Q'_{1\times n}$. In more detail, there exists $I_{1\times m} \in \mathbf{N}^m$ such that $\forall x \in \{1, 2, \ldots, n\}$, q_x-$q'_x = I_{1\times m}\bullet (k_{x1}, k_{x2}, \ldots, k_{xm})^T$. That is to say, there exists $C_{1\times m} = (c_1, c_2, \ldots, c_m) \in \mathbf{N}^m$ such that $\forall x \in \{1, 2, \ldots, n\}$, q_x-$q'_x = C_{1\times m}\bullet (k_{x1}, k_{x2}, \ldots, k_{xm})^T$ if S_{1x} and S_{2x} are two ω-numbers with the same form and $S_{1x} = S_{2x}$ if they are both integers. ■

Property 4 provides a criterion for the determination on whether there exists an inclusion relation between two ω-vectors with the same form.

Consider two ω-vectors with the same form $\mu_1 = (S_{11}, S_{12}, S_{13}) = (2\omega^{(1)} + \omega^{(2)}, \omega^{(2)}, 1)$ and $\mu_2 = (S_{21}, S_{22}, S_{23}) = (2\omega^{(1)} + \omega^{(2)} + 2, \omega^{(2)} + 1, 1)$.

For S_{11} and S_{21}, $2 = C_{1\times m} \bullet (2, 1)^T$ => $C_{1\times m} = (1, 0)$ or (0, 2);

For S_{12} and S_{22}, $1 = C_{1\times m} \bullet (0, 1)^T$ => $C_{1\times m} = (c, 1)$, where c can be any nonnegative integer; and $S_{23} = S_{13}$.

Clearly, there does not exist a same vector $C_{1\times m}$ for the first and second components of μ_1 and μ_2. Hence, $\mu_2 \not\subseteq \mu_1$.

Consider another pair of ω-vectors with the same form $\mu_3 = (S_{31}, S_{32}, S_{33}) = (2\omega^{(1)} + \omega^{(2)}, \omega^{(2)}, 1)$ and $\mu_4 = (S_{41}, S_{42}, S_{43}) = (2\omega^{(1)} + \omega^{(2)} + 2, \omega^{(2)} + 2, 1)$.

For S_{41} and S_{31}, 2= $C_{1\times m}\bullet$ $(2, 1)^T$ => $C_{1\times m} = (1, 0)$ or $(0, 2)$;

For S_{42} and S_{32}, 2= $C_{1\times m}\bullet$ $(0, 1)^T$ => $C_{1\times m} = (c, 2)$, where c can be any nonnegative integer; and $S_{43} = S_{33}$.

Since there exists a same vector $C_{1\times m} = (0, 2)$ for the first and second components, and the third components are a same integer, we have $\mu_4 \subseteq \mu_3$.

Note that whether there exists an inclusion relation between two independent ω-vectors with the same form can be determined by Properties 3 or 4. Obviously, such a determination via Property 3 is easier and thus is used.

IV. CONSTRUCTION OF NEW REACHABILITY TREES

A. ALGORITHM FOR NRT CONSTRUCTION

Based on modified ω-numbers and ω-vectors as well as their related notions, the construction algorithm of *new reachability trees* (NRTs) for unbounded PNs is developed in this section. Importantly, this algorithm is capable of checking whether or not the reachability set of an unbounded PN is semilinear. Before presenting the algorithm, some necessary concepts are introduced as follows.

Definition 11: Given two markings μ_1 and μ_2 with $\mu_2 > \mu_1$ and $\mu_2 \not\subseteq \mu_1$, and an ω-element $\omega^{(k)}$, we define $\mu_2' = \Gamma(\mu_1, \mu_2, \omega^{(k)})$ such that $\forall p \in P$, $\mu_2'(p) = \mu_2(p) + (\mu_2(p) - \mu_1(p)) \cdot \omega^{(k)}$.

For example, there are two markings $\mu_1 = (\omega^{(1)} + 1, 2, 0)$ and $\mu_2 = (\omega^{(1)} + 3, 2, 1)$, and an ω-element $\omega^{(2)}$. Clearly, $\mu_2 > \mu_1$ and $\mu_2 \not\subseteq \mu_1$. We have $\mu_2' = \Gamma(\mu_1, \mu_2, \omega^{(2)}) = (\omega^{(1)} + 2\omega^{(2)} + 3, 2, \omega^{(2)} + 1)$.

Besides, in the NRT construction algorithm, given an ω-marking μ, we also use an ω-element notation, e.g., $\omega^{(j)}$, to denote an ordinary vector where each entry is the base k_j related to $\omega^{(j)}$ in each entry of μ. For example, given a marking $\mu = (1, \omega^{(1)} + 2\omega^{(2)} + 4, 0, 0, \omega^{(1)} + \omega^{(2)} + 2)$, we have $\omega^{(1)} = (0, 1 ,0, 0, 1)$ and $\omega^{(2)} = (0, 2, 0, 0, 1)$.

In addition, the *next-state function* $\delta(\mu, t)$ is repeatedly called in the NRT construction algorithm, which computes the marking resulting from firing t once at the current marking μ. The detailed computation of $\delta(\mu, t)$ proceeds as follows [24].

Case 1: t is enabled at μ, where μ is an ordinary marking or ω-marking.

$\delta(\mu, \text{t})$ is computed according to the transition firing rule for ordinary markings, during which the addition of an ω-number and an integer is performed by Definition 4.

Case 2: t is conditionally enabled at μ, where μ is an ω-marking.

First, remove all ordinary markings of μ at which t is not enabled and denote the obtained ω-marking as μ'. Next, let $\delta(\mu, t) = \delta(\mu', t)$, where $\delta(\mu', t)$ is computed as Case 1.

To construct an NRT, we introduce four types of nodes, which are originally used in [24] to construct an NMRT. They are *terminal*, *duplicate*, *ω-duplicate* and *common* nodes depicted by ●, ⊖, ⦶, and ○, respectively. A terminal node is one that corresponds to a dead marking, i.e., a marking at which none of transitions is enabled. A duplicate node is one the same as a node that already appears in the tree along the same path. An ω-duplicate node is one with an ω-marking contained in the ω-marking of a node that appears already in the tree along the same path.

Now, we show the construction algorithm (Algorithm 1) of an NRT for unbounded nets.

Note that Algorithm 1 determines whether or not an unbounded PN has semilinear reachability set and it constructs a tree called a *new reachability tree* (NRT) in the case that the unbounded PN has semilinear reachability set. Now, we explain Algorithm 1 in more detial. We can see that a recursive function *GenerateSonNodes* is involved in Algorithm 1. By recursively calling *GenerateSonNodes*, nodes of the tree are created one after another according to the rule of depth-first search. When the created node is determined to be a terminal node, ω-duplicate node, or duplicate node, we do not generate its son nodes any more. Now, let us focus on how Algorithm 1 computes the marking of each created node, which is shown in Steps 3-17. In more detail, every time a transition t is enabled or conditionally enabled at a marking μ_x, we firstly create a new node z and compute the next-state $\delta(\mu_x, t)$. Next, we should determine *if there exists a path in the constructed tree from a node y to x such that* $\delta(\mu_x, t) > \mu_y$ *and* $\delta(\mu_x, t) \not\subseteq \mu_y$. Note that which ω-marking is bigger and whether there exists an inclusion relation between two ω-markings can be decided by Definition 10 and Properties 3 or 4, respectively. If such a path does not exist, the marking of node z is exactly $\delta(\mu_x, t)$. Otherwise, an ω-element with a new superscript k should be introduced to the marking of node z, resulting in $\mu_z := \Gamma(\mu_y, \delta(\mu_x, t), \omega^{(k)})$. It is worth noting that every time we introduce a new ω-element $\omega^{(k)}$, we accordingly compute $v^{(k)}$ as shown in Step 11. In addition, we determine whether the execution of Algorithm 1 should be aborted as shown in Steps 12-14. If so, the reachability set of the unbounded PN is determined to be not semilinear. Note that in Step 12, each ω-element in condition 1) denotes an ordinary vector as stated before. Here, we present the following result.

Proposition 1: Suppose that Algorithm 1 never exits in Step 13. Infinite ω-elements will be introduced in NRT if and only if the condition in Step 12 holds during the execution of Algorithm 1.

Proof (Sufficiency): Since $\omega^{(i)}$, $\omega^{(j)}$, $\omega^{(k)}$ are introduced in μ_z with $i < j < k$, the transition sequence $\sigma = \alpha_0\sigma_i\alpha_1\sigma_j\alpha_2\sigma_k \in T^*$ can fire from the initial marking μ_0 of the input net, where σ_i, σ_j, σ_k are the corresponding transition sequences that lead to the introduction of $\omega^{(i)}$, $\omega^{(j)}$, $\omega^{(k)}$, respectively. In other words, σ_i, σ_j, σ_k are transition sequences correspond to Parikh vectors $v^{(i)}$, $v^{(j)}$, $v^{(k)}$ computed in Step 11. Let $\mu_1, \mu_2, \ldots, \mu_6$ be ordinary markings

Algorithm 1 Semilinearity Checking of Reachability Sets and NRT Construction

Input: An unbounded PN (N, μ_0).
Output: 1) An NRT of (N, μ_0) and "semilinearity"; or 2) "non-semilinearity".

```
1.  Create a tree with only one node x_0, let μ_0 be the
    marking of node x_0 and label x_0 the root node;
2.  GenerateSonNodes(x_0, μ_0);
3.  Output "semilinearity".
Function GenerateSonNodes(x, μ_x)
Input: A node x and its corresponding marking μ_x.
1.  for each t ∈ T do
2.    if t is enabled or conditionally enabled at μ_x then
3.      Create a new node z and compute the next-state
        δ(μ_x, t);
4.      if there exists a path in the constructed tree from
        a node y to x such that δ(μ_x, t) > μ_y and δ(μ_x, t) ⊄ μ_y
        then
          /*Introduce a new superscript k that never appears
          in δ(μ_x, t)*/
5.        if δ(μ_x, t) is an ordinary marking then
6.          k:=1;
7.        else
8.          k:=1+g, where g is the maximal dimension
            of all the ω-numbers in δ(μ_x, t);
9.        end if
10.       μ_z:= Γ(μ_y, δ(μ_x, t), ω^(k));
11.       Let ν^(k) be the Parikh vector of the transition
          sequence σ_yx t, where σ_yx is the transition sequence
          associated with the path from y to x;
12.       if there exist ω^(i) and ω^(j) in μ_z with i < j < k
          such that 1) ω^(i) < ω^(j) < ω^(k); 2) ν^(i) < ν^(j) < ν^(k);
          and 3) ∃b ∈ Z^+ such that b·(ν^(j) − ν^(i)) = ν^(k) − ν^(j)
          then
13.         Exit and Output "non-semilinearity";
14.       end if
15.     else
16.       μ_z:= δ(μ_x, t);
17.     end if
        /* Add an arc from x to z*/
18.     if t is enabled at μ_x then
19.       Add a solid arc labeled t from x to z;
20.     else /* t is conditionally enabled at μ_x*/
21.       Add a dash arc labeled t from x to z;
22.     end if
23.     if z is a terminal node, ω-duplicate node, or
        duplicate node then
24.       Denote node z by the corresponding
          filled circle;
25.     else /* z is a common node*/
26.       GenerateSonNodes(z, μ_z);
27.     end if
28.   end if
29. end for
```

such that $\mu_0[\alpha_0\rangle\mu_1[\sigma_i\rangle\mu_2[\alpha_1\rangle\mu_3[\sigma_j\rangle\mu_4[\alpha_2\rangle\mu_5[\sigma_k\rangle\mu_6$. Since $\omega^{(i)} < \omega^{(j)} < \omega^{(k)}$, we have $\mu_2 - \mu_1 < \mu_4 - \mu_3 < \mu_6 - \mu_5$. It is trivial to see that $\mu_3 > \mu_1$, $\mu_5 > \mu_3$, and there exits $\alpha_3 \in T^*$ such that $\mu_6[\alpha_3\rangle\mu_7$ and $\mu_7 > \mu_5$. Clearly, $\mu_2 - \mu_1 = [N] \cdot \nu^{(i)}$, $\mu_4 - \mu_3 = [N] \cdot \nu^{(j)}$, and $\mu_6 - \mu_5 = [N] \cdot \nu^{(k)}$, where N is the input net. Since $\nu^{(i)} < \nu^{(j)} < \nu^{(k)}$ and $b \cdot (\nu^{(j)} - \nu^{(i)}) = \nu^{(k)} - \nu^{(j)}$, it holds $b \cdot (\omega^{(j)} - \omega^{(i)}) = \omega^{(k)} - \omega^{(j)}$. Let σ_l be a transition sequence with its Parikh vector $\nu^{(l)}$ satisfying that $\nu^{(k)} < \nu^{(l)}$ and $b \cdot (\nu^{(k)} - \nu^{(j)}) = \nu^{(l)} - \nu^{(k)}$. Since σ_i is enabled at μ_1, σ_j is enabled at μ_3, σ_k is enabled at μ_5, we can see that σ_l is enabled at μ_7, resulting in $\mu_7[\sigma_l\rangle\mu_8$. Furthermore, the ω-element $\omega^{(l)} = \mu_8 - \mu_7$ can be introduced in NRT. Clearly, $\mu_8 - \mu_7 = [N] \cdot \nu^{(l)}$. Trivially, we have $\omega^{(l)} > \omega^{(k)}$ and it holds $b \cdot (\omega^{(k)} - \omega^{(j)}) = \omega^{(l)} - \omega^{(k)}$. Now, since $\omega^{(j)} < \omega^{(k)} < \omega^{(l)}$, $\nu^{(j)} < \nu^{(k)} < \nu^{(l)}$ and $b \cdot (\nu^{(k)} - \nu^{(j)}) = \nu^{(l)} - \nu^{(k)}$, similarly, another ω-element can be introduced in NRT. By repeating the above reason, ω-elements with different superscripts are introduced infinitely.

(Necessity): It is trivial to see that there exists an infinitely long path in NRT where infinite ω-elements $\omega^{(1)}$, $\omega^{(2)}, \ldots, \omega^{(\infty)}$ are introduced one after another. Let σ_1, $\sigma_2, \ldots, \sigma_\infty$ be the corresponding transition sequences that lead to the introduction of $\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(\infty)}$, respectively. In other words, $\sigma_1, \sigma_2, \ldots, \sigma_\infty$ are transition sequences correspond to $\nu^{(1)}, \nu^{(2)}, \ldots, \nu^{(\infty)}$ that are computed in Step 11. Trivially, we can see that the infinitely long transition sequence $\sigma = \alpha_0\sigma_1\sigma_2\ldots\sigma_\infty$, where $\alpha_0 \in T^*$, can fire from the initial marking μ_0 of the input net. Since σ is infinitely long, there exist $i_1, i_2, \ldots, i_\infty \in \{1, 2, \ldots, \infty\}$ with $i_1 < i_2 < \ldots < i_\infty$ such that $\omega^{(i1)} < \omega^{(i2)} < \ldots < \omega^{(i\infty)}$ and $\nu^{(i1)} < \nu^{(i2)} < \ldots < \nu^{(i\infty)}$, where $\nu^{(i1)}$, $\nu^{(i2)}, \ldots, \nu^{(i\infty)}$ denote Parikh vectors of transition sequences $\sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{i\infty}$. Furthermore, due to the fact that transitions in a PN are limited and the characteristics of marking evolution with the firing of a transition sequence, it holds that $\exists b \in \mathbf{Z}^+$ such that $\nu^{(i(q+1))} - \nu^{(i(q))} = b^{q-1} \cdot \Delta$, where $q \in \{1, 2, \ldots, \infty\}$ and $\Delta = \nu^{(i2)} - \nu^{(i1)}$. Hence, the condition in Step 12 holds during the execution of Algorithm 1. ■

Remark 3: By Proposition 1, we actually determine in Step 12 whether infinite ω-elements with different superscripts will be introduced in the future steps of Algorithm 1. Indeed, if we ignore Steps 12-14, it could happen that ω-elements with different superscripts are introduced infinitely during the execution of Algorithm 1. As a result, Algorithm 1 cannot terminate in finite steps and clearly a finite NRT cannot be constructed. This is why Algorithm 1 includes Steps 12-14, by which the case of introducing infinite ω-elements in NRT is removed. Moreover, note that the marking of each node in NRT is actually a linear set. Hence, introducing infinite ω-elements in NRT implies that the reachability set of the handled PN is not semilinear since a semilinear set is a finite union of linear sets. As a result, Algorithm 1 is capable of determining whether the input unbounded PN has the semilinear reachability set. Besides, it constructs a finite NRT in the case that the reachability set of the input net is semilinear. Indeed, a wide variety of subclasses of PNs enjoy semilinear reachability sets. It is proved in [28] that persistent PNs, weakly persistent PNs,

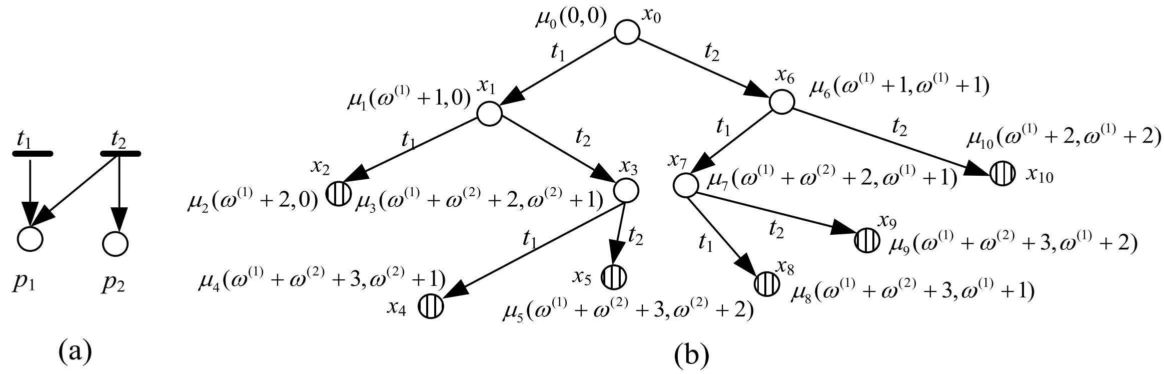


FIGURE 1. **(a) A simple unbounded PN and (b) its NRT.**

almost persistent PNs, sinkless PNs, almost sinkless PNs, and cyclic PNs all have semilinear reachability sets.

In the following, we present two examples to illustrate Algorithm 1.

Example 1: Consider the unbounded PN in Fig. 1(a) whose reachability set is semilinear. According to Algorithm 1, we construct its NRT as follows. First, creat the root node x_0 and let $\mu_0 = (0, 0)$ be the marking of x_0. Next, *GenerateSonNodes*(x_0, μ_0) is called to generate all son nodes of x_0 according to the rule of depth-first search: It can be seen that t_1 is enabled at μ_0. Hence, we create a new node x_1 and compute $\delta(\mu_0, t_1) = (1, 0)$. Clearly, $\delta(\mu_0, t_1) > \mu_0$ and $\delta(\mu_0, t_1) \not\subset \mu_0$. Hence, the ω-element $\omega^{(1)}$ is introduced, resulting in the marking of node x_1 being $\mu_1 = \Gamma(\mu_0, \delta(\mu_0, t_1), \omega^{(1)}) = (\omega^{(1)} + 1, 0)$. Note that $\nu^{(1)}$ should be computed, that is, $\nu^{(1)} = [1, 0]^T$. Since t_1 is enabled at μ_0, a solid arc labeled t_1 is added from x_0 to x_1. We can see x_1 is a common node. Hence, *GenerateSonNodes*(x_1, μ_1) is called to generate all its son nodes. Similarly, since t_1 is enabled at μ_1, we create a new node x_2 and compute $\delta(\mu_1, t_1) = (\omega^{(1)}+2, 0)$. Note that condition in Step 4 does not hold now and thus the marking of x_2 is exactly $\delta(\mu_1, t_1)$, i.e., $\mu_2 = (\omega^{(1)} + 2, 0)$. Node x_2 is an ω-duplicate node since $\mu_2 \subset \mu_1$. Now, we consider another transition t_2 that is enabled at μ_1. Similarly, a node x_3 is created with the marking $\mu_3 = (\omega^{(1)} + \omega^{(2)} + 2, \omega^{(2)} + 1)$. Note that $\nu^{(2)} = [0, 1]^T$ and the condition in Step 12 does not hold. Since x_3 is a common node, *GenerateSonNodes*(x_3, μ_3) is called to generate all its son nodes. By repeating the simliar way, other nodes with markings $\mu_4 = (\omega^{(1)}+\omega^{(2)}+3, \omega^{(2)} + 1)$, $\mu_5 = (\omega^{(1)} + \omega^{(2)} + 3, \omega^{(2)} + 2)$, $\mu_6 = (\omega^{(1)} + 1, \omega^{(1)}+1)$, $\mu_7 = (\omega^{(1)}+\omega^{(2)}+2, \omega^{(1)}+1)$, $\mu_8 = (\omega^{(1)}+\omega^{(2)}+3, \omega^{(1)}+1)$, $\mu_9 = (\omega^{(1)}+\omega^{(2)}+3, \omega^{(1)}+2)$ and $\mu_{10} = (\omega^{(1)}+2, \omega^{(1)} + 2)$ are generated one after another. Finally, we obtain the NRT, as depicted in Fig. 1(b). Note that the condition in Step 12 never holds during the execution of Algorithm 1 and Algorithm 1 also outputs "semilinearity" finally.

Example 2: Consider the unbounded PN in Fig. 2 whose reachability set is not semilinear [7]. Suppose that we apply a modified Algorithm 1 to the PN with Steps 12-14 removed. We can see that the modified Algorithm 1 cannot terminate since infinite ω-elements with different superscripts are

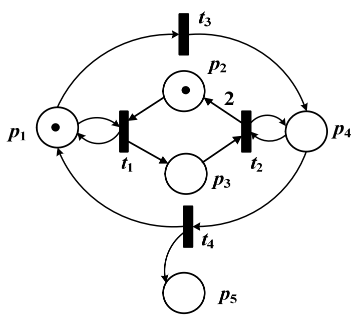


FIGURE 2. **An unbounded PN introduced in [7].**

introduced during the construction of the reachability tree. In more detail, it could happen that after firing $t_1t_3t_2t_4$ from the initial marking $\mu_0 = (1, 1, 0, 0, 0)$, a node with an ω-marking $\mu_1 = (1, \omega^{(1)} + 2, 0, 0, \omega^{(1)} + 1)$ is created in the reachability tree. Next, after firing $2t_1t_32t_2t_4$ from μ_1, a node with an ω-marking $\mu_2 = (1, \omega^{(1)} + 2\omega^{(2)} + 4, 0, 0, \omega^{(1)} + \omega^{(2)}+2)$ is created. By repeating the above procedure, we can know that after firing $t_1t_3t_2t_42t_1t_32t_2t_4 \ldots 2^{n-1}t_1t_32^{n-1}t_2t_4$ from μ_0, a node with an ω-marking $\mu_n = (1, \omega^{(1)} + 2\omega^{(2)} + \ldots + 2^{n-1}\omega^{(n)} + 2^n, 0, 0, \omega^{(1)}+\omega^{(2)}+\ldots+\omega^{(n)}+n)$ is created. Clearly, infinite ω-elements with different superscripts can be introduced since n can be infinite. Now, consider applying Algorithm 1 to the PN. We note that when the node with the ω-marking $\mu_1 = (1, \omega^{(1)}+2, 0, 0, \omega^{(1)}+1)$ is created, we have $\nu^{(1)} = [1\ 1\ 1\ 1]^T$; when the node with the ω-marking $\mu_2 = (1, \omega^{(1)} + 2\omega^{(2)} + 4, 0, 0, \omega^{(1)} + \omega^{(2)} + 2)$ is created, we have $\nu^{(2)} = [2\ 2\ 1\ 1]^T$; and when the node with the ω-marking $\mu_3 = (1, \omega^{(1)}+2\omega^{(2)}+4\omega^{(3)}+8, 0, 0, \omega^{(1)}+\omega^{(2)}+\omega^{(3)}+3)$ is created, we have $\nu^{(3)} = [4\ 4\ 1\ 1]^T$. We can see that $\nu^{(1)} < \nu^{(2)} < \nu^{(3)}$ and $2(\nu^{(2)}-\nu^{(1)}) = \nu^{(3)}-\nu^{(2)}$. Besides, note that $\omega^{(1)} = [0\ 1\ 0\ 0\ 1]^T$, $\omega^{(2)} = [0\ 2\ 0\ 0\ 1]^T$, and $\omega^{(3)} = [0\ 4\ 0\ 0\ 1]^T$. Clearly, $\omega^{(1)} < \omega^{(2)} < \omega^{(3)}$. Hence, Algorithm 1 exits and outputs "non-semilinearity".

B. RESULTS RELATED TO NRT

In this section, we formally prove that Algorithm 1 outputs a finite NRT in the case that the input unbound PN has semilinear reachability set and NRTs exactly characterize

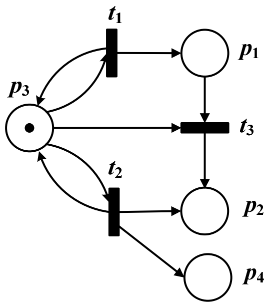


FIGURE 3. An unbounded net.

the reachability sets of unbounded PNs. Moreover, we show that whether unbounded PNs suffer from deadlocks can be checked based on NRTs.

Before showing that Algorithm 1 outputs finite NRTs, some lemmas about NRTs are revealed.

Lemma 1: Let $\mu_0, \mu_1, \mu_2, \ldots, \mu_n$ be a sequence of markings corresponding to a path starting from the root node of NRT. ω-numbers with the same dimension in each coordinate of $\mu_0, \mu_1, \mu_2, \ldots,$ and μ_n have the same form.

Proof: By Algorithm 1, once a new ω-element $\omega^{(j)}$ is introduced into a component of a marking, the base related to $\omega^{(j)}$ can never be changed in the corresponding component of any marking that is generated subsequently in the same path. Hence, the conclusion holds. ■

For example, there is a sequence of markings $\mu_0, \mu_1, \mu_3, \mu_5$ corresponds to a path in Fig. 1(b), whose first coordinate is $0, \omega^{(1)}+1, \omega^{(1)}+\omega^{(2)}+2, \omega^{(1)}+\omega^{(2)}+3$. Clearly, $\omega^{(1)}+\omega^{(2)}+2$ and $\omega^{(1)}+\omega^{(2)}+3$ are ω-numbers with two-dimension and they have the same form.

Lemma 2: Let $\mu_0, \mu_1, \mu_2, \ldots, \mu_n$ be a sequence of markings corresponding to a path starting from the root node of NRT. $u_j \subseteq u_i$ if $u_j > u_i$, where $i, j \in \{0, 1, 2, \ldots, n\}$ and $j > i$.

Proof: Since $u_j > u_i$, we know that u_j and u_i are ω-markings with the same form or both ordinary markings. According to the construction of NRT, it can be concluded that $u_j \subseteq u_i$ since otherwise a new superscript has to be introduced into u_j. ■

Lemma 3 [26]: In any infinite directed tree where each node has only a finite number of direct successors, there exists an infinite path leading from the root.

Theorem 1 (Finiteness): The NRT output by Algorithm 1 is finite.

Proof: By contradiction, suppose that there exists an infinite NRT. Due to Lemma 3, there exists an infinite path $x_0, x_1, x_2, \ldots$ from the root node x_0. Accordingly, we have an infinite sequence of makings, denoted as $u[x_0], u[x_1], u[x_2], \ldots$.

Consider the first coordinate of $u[x_0], u[x_1], u[x_2], \ldots,$ denoted as $u[x_0](p_1), u[x_1](p_1), u[x_2](p_1), \ldots$. By Proposition 1, it is impossible to introduce infinite superscripts during constructing NRT. Hence, there exists $u[x_a](p_1)$ in $u[x_0](p_1)$, $u[x_1](p_1), u[x_2](p_1), \ldots,$ which is an ω-number with the maximal dimension. Besides, it is easy to know that $u[x_a](p_1)$, $u[x_{a+1}](p_1), u[x_{a+2}](p_1), \ldots$ is an infinite sequence of ω-numbers with the same dimension. According to Lemma 1, $u[x_a](p_1), u[x_{a+1}](p_1), u[x_{a+2}](p_1), \ldots$ is an infinite sequence of ω-numbers with the same form. Hence, an infinite nondecreasing subsequence can be definitely found in $u[x_a](p_1), u[x_{a+1}](p_1), u[x_{a+2}](p_1), \ldots$. In other words, we can find an infinite node subsequence of $x_0, x_1, x_2, \ldots,$ denoted as $x'_0, x'_1, x'_2, \ldots$ such that $u[x'_0](p_1) \leq u[x'_1](p_1) \leq u[x'_2](p_1) \leq \ldots$. Now, consider the second coordinate of $u[x'_0], u[x'_1], u[x'_2], \ldots,$ denoted as $u[x'_0](p_2), u[x'_1](p_2), u[x'_2](p_2), \ldots$. Similarly, we can also find an infinite node subsequence of $x'_0, x'_1, x'_2, \ldots,$ denoted as $x''_0, x''_1, x''_2, \ldots,$ such that $u[x''_0](p_2) \leq u[x''_1](p_2) \leq u[x''_2](p_2) \leq \ldots$. Finally, we can definitely find an infinite node subsequence of $x_0, x_1, x_2, \ldots,$ denoted as $x^*_0, x^*_1, x^*_2, \ldots$ such that $u[x^*_0] \leq u[x^*_1] \leq u[x^*_2] \leq \ldots$. But by construction, it must be an infinite strictly increasing subsequence, i.e., $u[x^*_0] < u[x^*_1] < u[x^*_2] < \ldots,$ since otherwise a node would be a duplicate one with no successors. Besides, $\forall u[x^*_j] > u[x^*_i], u[x^*_j] \not\subseteq u[x^*_i]$, since otherwise a node would be a ω-duplicate one with no successors. However, this contradicts Lemma 2. Therefore, the NRT output by Algorithm 1 is finite. ■

Theorem 2 (Reachability): The NRT of an unbounded PN contains only but all reachable markings of the PN.

Proof: According to the construction algorithm of NRT, it is easy to conclude that the reachable marking set is contained in NRT. In what follows, we prove that any marking in NRT belongs to the reachable marking set. Firstly, consider the direct successors of the root node. Obviously, the marking sets corresponding to these direct successors are contained in the reachable marking set. Next, consider the director successors of a node whose marking set is contained in the reachable marking set. Similarly, it is easy to see that the marking sets of these director successors are also contained in the reachable marking set. As a result, we can conclude that any marking in NRT belongs to the reachable marking set. ■

Theorem 3 (Semilinear-Reachability-Set Checking): Given an unbounded PN, its reachability set is semilinear if and only if Algorithm 1 outputs an NRT with the net as the input.

Proof : Straightforward from Proposition 1, Theorems 1 and 2, as well as the definition of semilinear sets. ■

Here we introduce that a *full conditional node* is a node in an NRT with all its direct successors connected by dash arcs.

Theorem 4 (Deadlock Checking): An unbounded PN with semilinear reachability set has deadlocks if and only if its NRT contains terminal nodes or full conditional nodes.

Proof (Sufficiency): It is clear that a terminal node or a full conditional node definitely contains a dead marking. Hence, we can see that there is a dead marking in the NRT. By Theorem 2, the dead marking is reachable in the unbounded PN. As a result, the unbounded PN has deadlocks.

(Necessity): Since the unbounded PN has deadlocks, we can see that there is a dead marking in the NRT according to Theorem 2. Clearly, dead markings are only contained in terminal nodes or full conditional nodes in the NRT. As a

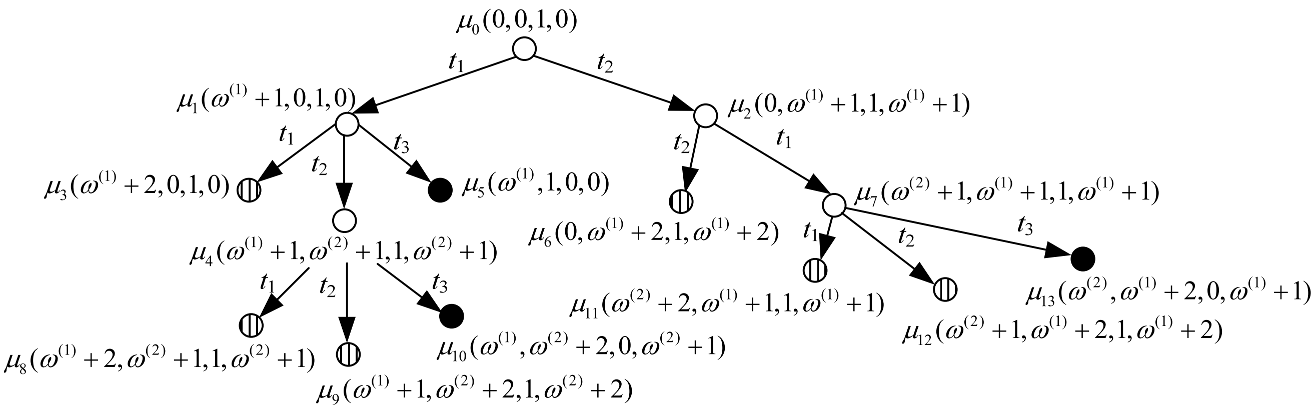


FIGURE 4. The NRT of the net in Figure 3.

result, the NRT contains terminal nodes or full conditional nodes. ■

For example, it is easy to decide that the unbounded PN in Fig. 1(a) suffers from no deadlocks due to Theorem 4.

Remark 4: For unbounded PNs with semilinear reachability sets, it is feasible to check their liveness based on NRTs. The basic idea of their liveness checking is similar to that in [22]. The specific checking procedure will be treated as our future work topic.

Example 3: Consider the unbounded PN in Fig. 3. According to Algorithm 1, its NRT is constructed as depicted in Fig. 4, which is clear a finite tree. Since Algorithm 1 outputs an NRT, the reachability set of the PN is semilinear. Besides, it can be checked that the NRT contains only but all reachable markings of the PN. According to Theorem 4, it can be determined that the PN in Fig. 3 suffers from deadlocks, which however cannot be decided via MRT or NMRT.

Finally, we point out that the computational complexity of Algorithm 1 is exponential with repsect to the size of the input PN [41],[42].

V. CONCLUSIONS AND FUTURE WORK

Finding a finite reachability tree to exactly represent the reachability set of an unbounded net is a rather complicated task. Only limited progress has been made in this respect since it was studied over half a century ago. This work presents an algorithm that is capable of determing whether an unbounded PN has the semilinear reachability set. Moreover, for unbounded PNs with semilinear reachability sets, the algorithm constructs finite new reachability trees (NRTs) where ω-numbers $\boldsymbol{\omega}(k_1^{(1)}, k_2^{(2)}, \ldots, k_m^{(m)}; q)$ are used to represent infinite components of an ω-marking rather than the symbol ω or the expression $k\omega_n + q$. NRTs thus provide more useful information than FRTs, MRTs, and NMRTs. Besides, it is proved that an NRT exactly characterizes the reachability set of the corresponding unbounded PN and whether an unbounded PN contains a deadlock can be correctly checked based on its NRT.

The future research include 1) proposing a specific liveness checking method based on NRTs; 2) simplifying the structure of NRTs while keeping their abilities of detecting deadlocks; and 3) developing CAD tools to facilitate the applications of NRTs in various fields [1]–[3], [13], [18], [29], [32], [34]–[40], [44]–[46].

VI. ACKNOWLEDGMENT

This work is an extended version of the conference paper "S. G. Wang, M. C. Zhou, M. D. Gan, D. You, and Y. Li, "New reachability trees for unbounded Petri nets," 2015 IEEE International Conference on Robotics and Automation, Seattle, USA, pp. 3862-3867, 2015."

REFERENCES

[1] Y. Chen and Z. Li, "On structural minimality of optimal supervisors for flexible manufacturing systems," *Automatica*, vol. 48, no. 10, pp. 2647–2656, 2012.

[2] Y. Chen, Z. Li, and K. Barkaoui, "Maximally permissive liveness-enforcing supervisor with lowest implementation cost for flexible manufacturing systems," *Inf. Sci.*, vol. 256, pp. 74–90, Jan. 2014.

[3] Y. Chen, Z. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, May 2015.

[4] Z. Ding, C. Jiang, and M. Zhou, "Deadlock checking for one-place unbounded Petri nets based on modified reachability trees," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 38, no. 3, pp. 881–883, Jun. 2008.

[5] S. Ginsburg and E. H. Spanier, "Semigroups, Presburger formulas, and languages," *Pacific J. Math.*, vol. 16, no. 2, pp. 285–296, 1966.

[6] D. Hauschildt, "Semilinearity of the reachability set is decidable for Petri nets," Ph.D. dissertation, Dept. Comput. Sci., Univ. Hamburg, Hamburg, Germany, 1990.

[7] J. E. Hopcroft and J.-J. Pansiot, "On the reachability problem for 5-dimensional vector addition systems," *Theor. Comput. Sci.*, vol. 8, no. 2, pp. 135–159, 1979.

[8] K. Hiraishi, "Reduced state space representation for unbounded vector state spaces," in *Proc. Int. Conf. Appl. Theory Petri Nets*. Berlin, Germany: Springer, 1996. 230-248.

[9] M. D. Jeng and M. Y. Peng, "On the liveness problem of 1-place-unbounded Petri nets," in *Proc. Int. Conf. Syst., Man, Cybern., Comput. Cybern. Simulation*, Orlando, FL, USA, Oct. 1997, pp. 3221–3226.

[10] M. D. Jeng and M. Y. Peng, "Augmented reachability trees for 1-place-unbounded generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 2, pp. 173–183, Mar. 1999.

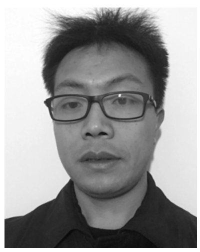
[11] R. M. Karp and R. E. Miller, "Parallel program schemata," *J. Comput. Syst. Sci.*, vol. 3, no. 2, pp. 147–195, 1969.

[12] J. L. Lambert, "Vector addition systems and semi-linearity," *SIAM J. Comput.*, 1994.

[13] J. Li, M. Zhou, T. Guo, Y. Gan, and X. Dai, "Robust control reconfiguration of resource allocation systems with Petri nets and integer programming," *Automatica*, vol. 50, no. 3, pp. 915–923, Mar. 2014.
[14] Y. Li, "Reachability trees for unbounded Petri nets," M.S. thesis, School Inf. Electron. Eng., Zhejiang Gongshang Univ., Hangzhou, China, 2014.
[15] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[16] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1981.
[17] Y. Ru, W. Wu, and C. N. Hadjicostis, "Comments on 'A modified reachability tree approach to analysis of unbounded Petri nets,'" *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 36, no. 5, p. 1210, Oct. 2006.
[18] M. Uzam, G. Gelen, and T. L. Saleh, "Think-globally-act-locally approach with weighted arcs to the synthesis of a liveness-enforcing supervisor for generalized Petri nets modeling FMSs," *Inf. Sci.*, vol. 363, pp. 235–260, Oct. 2016.
[19] F.-Y. Wang, "A modified reachability tree for Petri nets," in *Proc. Int. Conf. Syst., Man, Cybern.*, Charlottesville, VA, USA, Oct. 1991, pp. 329–334.
[20] F.-Y. Wang, Y. Gao, and M. Zhou, "A modified reachability tree approach to analysis of unbounded Petri nets," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 34, no. 1, pp. 303–308, Feb. 2004.
[21] H. M. Wong and M. C. Zhou, "Automated generation of modified reachability trees for Petri nets," in *Proc. Regional Control Conf.*, Brooklyn, NY, USA, 1992, pp. 119–121.
[22] S. G. Wang, M. D. Gan, and M. C. Zhou, "Macro liveness graph and liveness of ω-independent unbounded nets," *Sci. China Inf. Sci.*, vol. 58, no. 3, pp. 1–10, 2015.
[23] S. Wang, M. Gan, M. Zhou, and D. You, "A reduced reachability tree for a class of unbounded petri nets," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 4, pp. 345–352, Oct. 2015.
[24] S. G. Wang, M. Zhou, Z. Li, and C. Wang, "A new modified reachability tree approach and its applications to unbounded Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 43, no. 4, pp. 932–940, Jul. 2013.
[25] Y. Wang, B. Jiang, and L. Jiao, "Property checking for 1-place-unbounded Petri nets," in *Proc. IEEE Int. Symp. Theor. Aspects Softw. Eng. (TASE)*, Taipei, China, Aug. 2010, pp. 117–125.
[26] D. König, "Über eine Schlussweise aus dem Unendlichen ins Unendliche," *Acta Sci. Math. (Szeged)*, (in German), vol. 3, nos. 2–3, pp. 121–130, 1927.
[27] J. Leroux, "Presburger vector addition systems," in *Proc. 28th Annu. ACM/IEEE Symp. Logic Comput. Sci.*, Jun. 2013, pp. 23–32.
[28] H. Yen, "Path decomposition and semilinearity of Petri nets," *Int. J. Found. Comput. Sci.*, vol. 20, no. 4, pp. 581–596, 2009.
[29] G. Liu and C. Jiang, "Observable liveness of Petri nets with controllable and observable transitions," *Sci. China Inf. Sci.*, vol. 60, p. 118102, Nov. 2017.
[30] F. Lu *et al.*, "Complex reachability trees and their application to deadlock detection for unbounded Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2692262.
[31] R. Yang, Z. Ding, M. Pan, C. Jiang, and M. C. Zhou, "Liveness analysis of ω-independent Petri nets based on new modified reachability trees," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 9, pp. 2601–2612, Sep. 2017.
[32] Z. Ding, M. Jiang, H. Chen, Z. Jin and M. Zhou, "Petri net based test case generation for evolved specification," *Sci. China Inf. Sci.*, vol. 59, p. 080105, Aug. 2016.
[33] S. Wang, M. Zhou, M. Gan, D. You, and Y. Li "New reachability trees for unbounded Petri nets," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seattle, WA, USA, May 2015, pp. 3862–3867.
[34] S. Wang, D. You, M. Zhou, and C. Seatzu, "Characterization of admissible marking sets in Petri nets with uncontrollable transitions," *IEEE Trans. Autom. Control*, vol. 61, no. 7, pp. 1953–1958, Jul. 2016.
[35] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S 4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.
[36] S. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1403–1410, Aug. 2018.
[37] D. You, S. Wang, and M. Zhou, "Computation of strict minimal siphons in a class of Petri nets based on problem decomposition," *Inf. Sci.*, vol. 409–410, pp. 87–100, Oct. 017.
[38] D. You, S. Wang, Z. Li, and C. Wang, "Computation of an optimal transformed linear constraint in a class of Petri nets with uncontrollable transitions," *IEEE Access*, vol. 5, pp. 6780–6790, 2017.
[39] D. You, S. Wang, W. Dai, W. Wu, and Y. Jia, "An approach for enumerating minimal siphons in a subclass of Petri nets," *IEEE Access*, vol. 6, pp. 4255–4265, 2018.
[40] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li, "Decentralized diagnosis by Petri nets and integer linear programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2726108.
[41] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Inf. Sci.*, vol. 363, pp. 190–197, Oct. 2016.
[42] G. Liu, "Some complexity results for the soundness problem of workflow nets," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 322–328, Apr./Jun. 2014.
[43] S. G. Wang, D. You, and M. C. Zhou, "New reachability trees for analyzing unbounded Petri nets with semilinear reachability sets," *Sci. China Inf. Sci.*, to be published, doi: 10.1007/s11432-017-9446-3.
[44] H. Zhang, L. Feng, N. Wu, and Z. Li, "Integration of learning-based testing and supervisory control for requirements conformance of black-box reactive systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 2–15, Jan. 2018.
[45] H. Zhang, L. Feng, and Z. Li, "A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2018.2793662.
[46] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2762823.

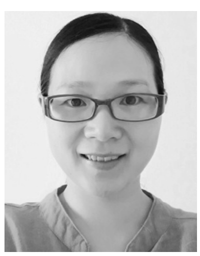
DAN YOU received the B.S. and M.S. degrees from the School of Information and Electronic Engineering, Zhejiang Gongshang University, China, in 2014 and 2017, respectively. She is currently a Member of the Discrete-Event System Group, School of Information and Electronic Engineering, Zhejiang Gongshang University. Her research interests include the supervisory control of discrete event systems and Petri net theory and application.

She is the author or coauthor of papers published in the IEEE Transactions on Automatic Control, *Information Sciences*, the IEEE Transactions on Systems, Man, Cybernetics: Systems, the IEEE Conference on Decision and Control, and the IEEE International Conference on Robotics and Automation.



SHOUGUANG WANG (M'10–SM'12) received the B.S. degree in computer science from the Changsha University of Science and Technology, Changsha, China, in 2000, and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005. He joined Zhejiang Gongshang University in 2005, where he is currently a Professor with the School of Information and Electronic Engineering, the Director of the Discrete-Event Systems Group and the Dean of the System Modeling and Control Research Institute, Zhejiang Gongshang University.

He was a Visiting Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, from 2011 to 2012. He was the Dean of the Department of Measuring & Control Technology and Instrument from 2011 to 2014. He was a Visiting Professor with the Electrical and Electronic Engineering Department, University of Cagliari, Cagliari, Italy, from 2014 to 2015.



WENHUI WU received the B.S. and M.S. degrees from Zhejiang Normal University, China, in 2003 and 2006, respectively. She is currently a Senior Lab Master with the School of Information and Electronic Engineering, Zhejiang Gongshang University. Her main interests include Petri net theory and application.