On Compositionality of Boundedness and Liveness for Nested Petri Nets

Leonid V. Dvoryansky*
National Research University Higher School of Economics,
Moscow, 101000, Russia
leo@mathtech.ru

Irina A. Lomazova*†
National Research University Higher School of Economics,
Moscow, 101000, Russia
Program Systems Institute of the Russian Academy of Science,
Pereslavl-Zalessky, 152020, Russia
i_lomazova@mail.ru

Abstract. Nested Petri nets (NP-nets) are Petri nets with net tokens. The liveness and boundedness problems are undecidable for two-level NP-nets [14]. Boundedness is in EXPSPACE and liveness is in EXPSPACE or worse for plain Petri nets [6]. However, for some restricted classes, e.g. for plain free-choice Petri nets, problems become more amenable to analysis. There is a polynomial time algorithm to check if a free-choice Petri net is live and bounded [4].

In this paper we prove, that for NP-nets boundedness can be checked in a compositional way, and define restrictions, under which liveness is also compositional. These results give a base to establish boundedness and liveness for NP-nets by checking these properties for separate plain components, which can belong to tractable Petri net subclasses, or be small enough for model checking.

Address for correspondence: i_lomazova@mail.ru
*The research is partially supported by the Russian Fund for Basic Research (projects 11-01-00737, 11-07-00549).
†This study was carried out within the National Research University Higher School of Economics’ Academic Fund Program in 2012-2013, research grant No. 11-01-0032.
1. Introduction

Nested Petri nets (NP-nets) [14] are an extension of high-level Petri nets according to the “nets-within-nets” approach. Nets in nets are extensively studied in the Petri net literature, as a formalism for modeling active objects, mobility and dynamics in distributed systems [1, 7, 9, 10, 16].

NP-nets define tokens as nets on the base of colored Petri nets [8], where a color is a marked net token. The behavior of a NP-net is defined according to the value semantics, when each net token represents an independent object.

NP-nets are a convenient formalism for modeling systems of dynamic interacting agents: an agent is represented by a net token, while agents are distributed in a system net. Levels in NP-nets are coordinated via synchronized transitions (simultaneous firing of transitions in adjacent levels of the model). Being less than Turing-powerful NP-nets are strictly more expressive than Petri nets. The liveness and boundedness problems are undecidable for two-level NP-nets [14]. In this paper we investigate subclasses of NP-nets, for which boundedness and liveness problems can be practically validated.

The liveness problem for another “nets-within-nets” formalism of Elementary Object Systems is studied in [11, 12]. It is proved there, that liveness and reachability are undecidable for general Elementary Object Systems and PSPACE-complete for safe Elementary Object Systems.

Though boundedness and liveness are decidable for plain Petri nets, these problems are not tractable for plain nets in general. Boundedness is in \( \text{EXPSPACE} \) and liveness is in \( \text{EXPSPACE} \) or worse for Petri nets [6]. So, analysis of Petri nets subclasses, for which these problems can be decided efficiently, is of great importance. An example of such subclasses is the class of Free-choice Petri nets. The property of well-formedness (liveness and boundedness) can be checked for free-choice Petri nets in polynomial time [4].

In this paper we study conditions, under which boundedness and liveness can be established for a NP-net in a compositional way, i.e. can be deduced from boundedness or, correspondingly, liveness for NP-net components, considered as plain Petri nets.

The obtained results can be applied e.g. to NP-nets with free-choice components and not more than one synchronization label. For such nets we thus get a technique for deriving boundedness and liveness in polynomial time.

Among approaches related to modular verification and property preservation for distributed systems are preservation of temporal logic properties for an automata-based formalism [3] and property preserving composition of Elementary Net systems [2].

The paper is organized as follows. In Section 2 we provide a motivating example of a simple P2P protocol, modeled as a NP-net with a single synchronization label. The formal definition of NP-nets and its execution semantics is given in Section 3. Compositionality of boundedness for NP-nets is the subject of Section 4. In Section 5 we consider liveness and its compositionality for a subclass of NP-nets with a single synchronization label. Section 6 concerns an extension of these results for NP-nets with several synchronization labels. Section 7 contains some conclusions.

2. Motivating example

In this section we give an example of a NP-net \( NPN \), modeling a core part of a P2P protocol.

A NP-net consists of a high-level system net with net tokens and element nets, which describe the
structure of net tokens. Every place in a system net is typed by an element net, so that a place labeled with an element net EN may contain a multi-set of marked EN as tokens.

In our example NP-net NP (Fig.1) consists of a system net SN which models the interaction protocol itself and element nets which model behavior of seeds. $E_1$ providing datum, peers $E_2$ seeking for datum and pipes $E_3$ — channels for a secure data transfer.

Seeds and peers are modeled in a rough way, since we are interested only in their interactions. A seed $E_1$ can be in one of “ready to upload” $p_1$, “uploading” $p_2$ and “reinitialization” $p_3$ states. Seed events are “upload started” $t_1$, “upload complete” $t_2$ and “reinitialization complete” $t_3$. A peer $E_2$ can be in “ready to download” $p_4$, or in “downloading” $p_5$ states. Peer events are “download started” $t_4$ and “download complete” $t_5$.

A pipe is a connection which can be used to change/improve such aspects of a data transfer as security, reliability etc. A pipe $E_3$ has seed ($p_6,p_7$) and peer ($p_{12},p_{13}$) interfaces and a 2-cells buffer ($p_8,p_9,p_{10},p_{11}$). Initially the pipe is waiting ($p_{15}$) when a seed and a peer become involved into interaction ($t_{11}$). The seed can start ($t_6$) transferring data ($p_6$) to the buffer and the peer starts waiting for the datum portion. When a transmission ($t_7$) is finished, the seed returns to the wait state ($p_7$) and the first cell of buffer is filled ($p_8$). Then a data portion propagates ($t_8$) to the second cell of the buffer, thus the first cell becomes empty ($p_9$) and the second cell becomes filled ($p_{10}$). When the peer has downloaded ($t_9$) the data portion from the second cell ($p_{10}$), the second buffer cell becomes empty ($p_{11}$), and the peer waits for another portion of data ($p_{13}$). When the seed has sent all datum, the pipe waits for a data portion to be propagated through the buffer to the peer ($p_{13}$) and then moves ($t_{12}$) to reinitialization state ($p_{14}$). When reinitialization is finished ($t_{10}$) the pipe completes a transfer cycle and becomes ready for another cycle ($p_{15}$). Now the next transfer cycle can start ($t_{11}$).

The system includes initial and final pools for seeds, peers and pipes. In the initial pool an agent is ready for interaction, while in the final pool an agent reinitializes its inner structure for the next interaction activity. There is no final pool for peers, since we do not model peer initialization routines. The initial pools for seeds ($p_{16}$), peers ($p_{17}$) and available pipes ($p_{18}$) are filled with some agents in the initial state of the system.

All four transitions in the system net are labeled with a synchronization label $\lambda$. It means, that such a transition can fire only simultaneously with transitions labeled with $\lambda$ in all net tokens involved in this firing. It means that, if there are a seed and a peer which are ready to transfer data and there is an available pipe (a transition labeled with $\lambda$ is enabled in each of the three involved net tokens), then they can start ($t_{11}$) interaction ($p_{19},p_{20},p_{21}$) controlled by the pipe net ($E_3$). After the seed and the peer have finished ($t_{16}$) their data exchange, the peer returns to the initial pool ($p_{17}$), the seed and the session get to their final pools ($p_{22},p_{23}$). After reinitializing a pipe returns ($t_{17}$) to the initial pool of pipes ($p_{18}$) and a peer returns ($t_{15}$) to its initial pool ($p_{16}$).

This is certainly only the core of the protocol, since such aspects as system termination, or exceptions (transmission failures) are not specified.

### 3. Nested Petri nets

In Section 2 we showed how NP-nets can be used to model interaction of distributed agents. In this section we give formal definitions of nested Petri nets (NP-nets).

Nested nets form an extension of colored nets over a special universe. So, we start by defining
Figure 1. A nested Petri net model of P2P protocol.
colored Petri nets parameterized with a value universe $U$.

First we briefly introduce the notation that will be used in the remainder of the paper. By $\mathbb{N}$ we denote the set of natural numbers including zero. For a set $S$ a bag (multiset) $m$ over $S$ is a mapping $m : S \to \mathbb{N}$. The set of all bags over $S$ is also denoted by $\mathbb{N}^S$. We use $+$ and $-$ for the sum and the difference of two bags and $=, <, >, \leq, \geq$ for comparisons of bags, which are defined in the standard way. We overload the set notation, writing $\emptyset$ for the empty bag and $\in$ for the element inclusion.

### 3.1. Colored Petri nets

Our definition of colored Petri nets does not essentially differ from the classical one [8], but it has been adapted for our purpose, notably by the addition of transition labels.

In a colored net each place is mapped to a type, which is a subset of $U$. We assume a language $Expr$ for arcs expressions over a set $Var$ of variables and a set $Con$ of constants with some fixed interpretation $\mathcal{I}$, s.t. for any type-consistent evaluation $\nu : Var \to U$ the value $\mathcal{I}(e, \nu) \in \mathbb{N}^U$ of an expression $e \in Expr$ is defined. We also assume a set $Lab$ of labels for transitions such that $\tau \notin Lab$. The label $\tau$ is the special “silent” label, while labels from $Lab$ mark externally observable firings.

**Definition 3.1.** A colored net over the universe $U$ is a 6-tuple $(P, T, F, \nu, \gamma, \Lambda)$, where

- $P$ and $T$ are disjoint finite sets of places, respectively transitions;
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $\nu : P \to 2^U$ is a place typing function, mapping $P$ to subsets of $U$;
- $\gamma : F \to Expr$ is an arc labeling function;
- $\Lambda : T \to Lab \cup \{\tau\}$ is a transition labeling function.

For an element $x \in P \cup T$ an arc $(y, x)$ is called an input arc, and an arc $(x, y)$ an output arc; a preset $\cdot x$ and a postset $x^\ast$ are subsets of $P \cup T$ such that $\cdot x = \{y | (y, x) \in F\}$ and $x^\ast = \{y | (x, y) \in F\}$.

Given a colored net $N = (P, T, F, \nu, \gamma, \Lambda)$ over the universe $U$, a marking of $N$ is a function $m : P \to \mathbb{N}^U$, s.t. $m(p) \in \mathbb{N}^{\nu(p)}$ for $p \in P$.

A pair $(N, m)$ of a Petri net and a marking is called a marked Petri net.

Let $N = (P, T, F, \nu, \gamma, \Lambda)$ be a colored Petri net. A transition $t \in T$ is enabled in a marking $m$ iff $\exists \nu \forall p \in P : (p, t) \in F \Rightarrow m(p) \geq \mathcal{I}(\gamma(p, t), \nu)$. Here $\nu : Var \to U$ is a variable evaluation, called also a binding. An enabled transition $t$ may fire yielding a new marking $m'(p) = m(p) - \mathcal{I}(\gamma(p, t), \nu) + \mathcal{I}(\tau(t, p), \nu)$ for each $p \in P$ (denoted $m \xrightarrow{t} m'$). The set of all markings reachable from a marking $m$ (via a sequence of firings) is denoted by $\mathcal{R}(m)$.

As usual, a marked colored net defines a transition system which represents the observable behavior of the net.

### 3.2. Nested Petri nets

Now we define nested Petri nets (NP-nets) — extended colored PNs over a special universe. This universe consists of elements of some finite set $S$ (called atomic tokens) and marked Petri nets (called net
tokens). For simplicity we consider here only two-level NP-nets, where net tokens are usual colored Petri nets.

Let \( S \) be a finite set of atomic objects. For a colored PN \( N \) by \( M(N,S) \) we denote the set of all marked nets, obtained from \( N \) by adding markings over the universe \( S \). Let then \( N_1, \ldots, N_k \) be colored PNs over the universe \( S \). Define a universe \( \mathcal{U}(N_1, \ldots, N_k) = S \cup M(N_1,S) \cup \cdots \cup M(N_k, S) \) with types \( S, M(N_1, S), \ldots, M(N_k, S) \). We denote \( \Omega(N_1, \ldots, N_k) = \{S, M(N_1, S), \ldots, M(N_k, S)\} \). By abuse of notation we say, that a place \( p \) with a type \( M(N, S) \) is typed by \( N \).

**Definition 3.2.** Let \( \text{Lab} \) be a set of transition labels and let \( N_1, \ldots, N_k \) be colored PNs over the universe \( S \), where all transitions are labeled with labels from \( \text{Lab} \cup \{\tau\} \).

A NP-net is a tuple \( NP = \langle N_1, \ldots, N_k, SN \rangle \), where \( N_1, \ldots, N_k \) are called element nets, and \( SN \) is called a system net. A system net \( SN = \langle P_{SN}, T_{SN}, F_{SN}, v, \gamma, \Lambda \rangle \) is a colored PN over the universe \( \mathcal{U}(N_1, \ldots, N_k) \), where places are typed by elements of \( \Omega = \Omega(N_1, \ldots, N_k) \), transition labels are from \( \text{Lab} \cup \{\tau\} \), and an arc expression language \( \text{Expr} \) is defined as follows.

Let \( \text{Con} \) be a set of constants interpreted over \( \mathcal{U} \) and \( \text{Var} \) a set of variables, typed with \( \Omega \)-types. Then an expression in \( \text{Expr} \) is a multiset of elements over \( Con \cup Var \) of the same type with two additional restrictions: for every transition \( t \in T_{SN} \)

1. constants or multiple instances of the same variable are not allowed in input arc expressions of \( t \);
2. each variable in an output arc expression for \( t \) occurs in one of the input arc expressions of \( t \).

Note that removing the first restriction on system net arc expressions makes NP-nets Turing-powerful [13], since without this restriction there would be a possibility to check, whether inner markings of two tokens in a current markings are equal, and hence to make a zero-test. The second restriction excludes infinite branching in a transition system, representing a behavior of a NP-net.

The interpretation of constants from \( \text{Con} \) is extended to the interpretation \( \mathcal{I} \) of expressions under a given binding of variables in the standard way.

We call a marked element net a net token, and an element from \( S \) an atomic token.

A marking in a NP-net is defined as a marking in its system net. So, a marking \( m : P_{SN} \rightarrow \mathbb{N}^\mathcal{U} \) in a NP-net maps each place in its system net to a multiset of element tokens or net tokens of appropriate type.

A behavior of a NP-net is composed of three kinds of steps (firings).

An element-autonomous step is the firing of a transition \( t \), labeled with \( \tau \), in one of the net tokens of the current marking according to the usual firing rule for colored Petri nets. Formally, let \( m \) be a marking in a NP-net \( NP, \alpha = (N, \mu) \in m(p) \) — a net token (colored Petri net) residing in the place \( p \in P_{SN} \) in \( m \). Let also \( t \) be enabled in \( \alpha \) and \( \mu \xrightarrow{t} \mu' \) in \( \alpha \). Then the element-autonomous step \( s = ([t[\alpha]]) \) is enabled in \( m \) and the result of \( s \)-firing is the new marking \( m' \), s.t. for all \( p' \in P_{SN} \setminus p \) : \( m'(p') = m(p') \), and \( m'(p) = m(p) - \alpha + (N, \mu') \). Note, that such step changes only the inner marking in one of the net tokens.

A system-autonomous step is the firing of a transition \( t \in T_{SN} \), labeled with \( \tau \), in the system net according to the firing rule for colored Petri nets, as if net tokens were just colored tokens without inner marking. Formally, the system-autonomous step \( s = (t) \) is enabled in a marking \( m \) iff there exists a binding \( \nu : \text{Var} \rightarrow \mathcal{U} \), s.t. \( \forall p \in P_{SN} : (p,t) \in F_{SN} \Rightarrow m(p) \geq I(\gamma(p,t), \nu) \). the result of \( s \)-firing is the new marking \( m'(p) = m(p) - I(\gamma(p,t), \nu) + I(\gamma(t,p), \nu) \) for each \( p \in P_{SN} \) (denoted \( m \xrightarrow{(t)} m' \).
An autonomous step in a system net can move, copy, generate, or remove tokens involved in the step, but doesn’t change their inner markings.

A (vertical) synchronization step is the simultaneous firing of a transition \( t \in T_{SN} \), labeled with some \( \lambda \in Lab \), in the system net together with firings of transitions \( t_1, \ldots, t_q (q \geq 1) \) also labeled with \( \lambda \), in all net tokens involved in (i.e. consumed by) this system net transition firing.

Formally, let \( m \) be a marking in a NP-net \( NP \), let \( t \in T_{SN} \) be labeled with \( \lambda \) and enabled in \( m \) via binding \( \nu \) as a system-autonomous step. We say that a net token \( \alpha \in m \) is involved in \( t \)-firing via binding \( \nu \) iff \( \alpha \in I(\gamma(p,t),\nu) \) for some \( p \in \cdot \). Let then \( \alpha_1 = (N_{i_1},\mu_1), \ldots, \alpha_q = (N_{i_q},\mu_q) \) be all net tokens involved in the firing of \( t \) via binding \( \nu \), and for each \( 1 \leq j \leq q \) there is a transition \( t_j \), labeled with \( \lambda \) in \( N_{i_j} \), such that \( t_j \) is enabled in a marking \( \mu_j \), and \( \mu_j \stackrel{t_j}{\rightarrow} \mu_j' \) in \( N_{i_j} \). Then the synchronization step \( s = (t,t_1,\ldots,t_q) \) is enabled in \( m \) for \( NP \), and the result of \( s \)-firing is the new marking \( m' \) defined as follows. For each \( p \in P_{SN} \): \( m'(p) = m(p) - I(\gamma(p,t),\nu) + I(\gamma(t,p),\nu') \), where for a variable \( x \)

\[
\nu'(x) = \begin{cases} 
(N,\mu'), & \text{if } x \text{ occurs in some input expression } \gamma(p,t) \text{ and } \nu(x) = (N,\mu), \\
\nu(x), & \text{if } x \text{ does not occur in any input expression for } t. 
\end{cases}
\]

Figure 2 gives an example of a synchronization step. The left part of this picture shows a marked fragment of a system net. Here a transition \( t \) has two input places \( p_1 \) and \( p_2 \), and two output places \( p_3 \) and \( p_4 \). In a current marking the place \( p_1 \) contains three net tokens, two of them, \( \alpha_1 \) and \( \alpha_2 \), are explicitly depicted. The place \( p_2 \) contains two net tokens, the structure and the marking of one of them is shown in the picture. Only the synchronization step is allowed here, since all transitions are labeled with the synchronization label \( \lambda \). A possible binding for variables \( x,y,z \) in the input arc expressions is \( x = \alpha_1, y = \alpha_2 \) and \( z = \alpha_3 \). Then the transitions \( t \) in the system net, \( t_1 \) in \( \alpha_1 \), \( t_1 \) in \( \alpha_3 \), and \( t_2 \) in \( \alpha_2 \) fire simultaneously. The resulting marking \( m' \) is shown on the right side of the picture. According to the output arc inscriptions two copies of \( \alpha_1 \) after \( t_1 \)-firing appear in \( p_3 \), the net token \( \alpha_3 \) disappears, \( \alpha_2 \) with a new marking is transported into the place \( p_4 \), and a new net \( \alpha_c \) appears in \( p_4 \) being a value of \( c \).

So, a transition, labeled with \( \lambda \in Lab \), in a system net consumes net tokens with enabled transitions, labeled with \( \lambda \). To exclude trivially dead transitions we add to our definition of NP-nets the following syntactical restriction: for each system net transition \( t \) labeled with \( \lambda \neq \tau \), for each \( p \in \cdot \) the type of \( p \) is an element net, containing at least one transition labeled with \( \lambda \).

Thus a step is a set of transitions (an one-element set in the case of an autonomous step). We write \( m \stackrel{s}{\rightarrow} m' \) for a step \( s \) converting a marking \( m \) into a marking \( m' \). By \( \text{Steps}(NP) \) we denote the set of all (potential) steps in \( NP \).

A run in a NP-net \( NP \) is a sequence \( \rho = m_0 \stackrel{s_1}{\rightarrow} m_1 \stackrel{s_2}{\rightarrow} \ldots \), where \( m_0,m_1,\ldots \) are markings, \( m_0 \) is an initial marking, and \( s_1,s_2,\ldots \) are steps in \( NP \). For a sequence of steps \( \sigma = s_1,\ldots,s_n \) we write \( m \stackrel{\sigma}{\rightarrow} m' \), and say that \( m' \) is reachable from \( m \), when \( m = m_0 \stackrel{s_1}{\rightarrow} m_1 \ldots \stackrel{s_n}{\rightarrow} m_n = m' \). By \( \mathcal{R}(NP,m) \) we denote the set of all markings reachable from \( m \) in \( NP \), and by abuse of notations we write \( \mathcal{R}(NP) \) for the set of all marking reachable in \( NP \) from its initial marking.

In what follows, when we speak of separate components of a NP-net, we consider a system net and element nets as usual colored Petri nets with autonomous firings. That means that net tokens in a system net marking are treated as plain tokens without inner markings, and synchronization labels are not taken into account. For a system net \( SN \) in a NP-net \( NP \) by \( SN \) we denote the colored Petri net, obtained from \( SN \) by changing all transition labels to \( \tau \). Similarly, for an element net \( N \) by \( \hat{N} \) we denote the colored Petri net, obtained from \( N \) by changing all transition labels to \( \tau \). Semantics for \( SN \) and \( \hat{N} \) is defined.
Let then \( m \) be a marking in a NP-net \( NP \), i.e. a mapping \( m : P_{SN} \rightarrow \mathbb{N}^\mathcal{U} \), where \( \mathcal{U} = S \cup \mathcal{M}(N_1, S) \cup \cdots \cup \mathcal{M}(N_k, S) \). By \( \hat{m} \) we denote a marking in \( \hat{S}N \) s.t. for each \( p \in P_{SN} \) we have \( \hat{m}(p) = m(p) \), if \( p \) is typed by \( S \), and for \( p \) typed with \( N_i \) we have \( \hat{m}(p) = c_p \cdot (N_i, 0) \), where \( (N_i, 0) \) is \( N_i \) with empty marking, and \( c_p \) is the number of tokens in \( m(p) \).

Note that net tokens of the same type (i.e., with the same net structure) are not distinguished in a system net autonomous firings. This follows from the first input arc expressions restriction for NP-nets, which eliminates comparing inner markings of net tokens. Moreover, since all tokens in a system net place are of the same type, enabledness of an autonomous transition in a system net depends only on numbers of tokens in its input places, and a system net considered as a separate component is actually equivalent to a p/t-net.

For further details on NP-nets see [13, 14]. Note, however, that here we consider a typed variant of NP-nets, where a type is instantiated to each place.
4. Boundedness of NP-nets

A place $p$ in a marked Petri net is called bounded iff for every reachable marking the number of tokens residing in $p$ does not exceed some fixed bound $\kappa \in \mathbb{N}$. A marked Petri net $PN$ is bounded iff all its places are bounded. A Petri net $PN$ is structurally bounded iff it is bounded for an arbitrary initial marking. It is also well known, that the boundedness of a given marked Petri net $(N, m_0)$ is equivalent to the finiteness of its reachability set $\mathcal{R}(PN, m_0)$. It is easy to extend both variants of this definition to NP-nets.

**Definition 4.1.** A marked NP-net $NP$ with an initial marking $m_0$ is $\kappa$-bounded ($\kappa \in \mathbb{N}$), iff for any reachable marking $m \in \mathcal{R}(PN, m_0)$ the number of tokens in any place of the system net $SN$ and any place of any net-token in $m$ is not greater than $\kappa$. A marked NP-net $NP$ is bounded, iff $NP$ is $\kappa$-bounded for some $\kappa$.

It is easy to note, that a marked NP-net $NP$ is bounded iff its reachability set $\mathcal{R}(PN, m_0)$ is finite, as for plain Petri nets.

It was proved in [13] that the Boundedness problem is undecidable for two-level NP-nets. The coverability tree construction, which in theory can be used for checking boundedness in the case of plain Petri nets, is not applicable for NP-nets.

The next theorem gives sufficient conditions for establishing boundedness for NP-nets in a compositional way.

**Theorem 4.1.** Let $NP = (SN, N_1, \ldots, N_k)$ be a marked NP-net with an initial marking $m_0$. If the system net in $NP$, all net-tokens in $m_0$, as well as all net constants in arc expressions in $NP$, considered as separate components, are bounded, then $NP$ is bounded.

**Proof:**

The proof of the theorem is rather straightforward. Let the system net, the net tokens and the net constants in the initial marking for $NP$ be $\kappa$-bounded.

Consider a reachable state $m$ in $NP$. It is easy to see, that $\hat{m}$ is reachable in $\widehat{SN}$ with the initial marking $m_0$. Indeed, let $m_0 \xrightarrow{\alpha} m_1 \ldots \xrightarrow{\alpha} m_n = m$, and denote by $s^0$ the projection of $s$ to transitions in $SN$, i.e. $s^{0} = t$, if $t \in T_{SN}$ and $t$ is a component of $s$, and $s^0 = \epsilon$, if $s$ is an element-autonomous step. Then $\hat{m_0} \xrightarrow{s^0} \hat{m_1} \ldots \xrightarrow{s^0} \hat{m_n} = \hat{m}$ is a run in $\widehat{SN}$, possibly including some number of stuttering steps $\epsilon$. Since $\widehat{SN}$ is $\kappa$-bounded, and $m(p)$ and $\hat{m}(p)$ contain the same number of tokens for each $p \in P_{SN}$, we get $|m(p)| \leq \kappa$.

Now let $\alpha = (N, \mu)$ be a net token in $m$ residing in a place $p \in P_{SN}$. Construct a run $\rho$ in $\widehat{N}$ by refining the run $m_0 \xrightarrow{\alpha} m_1 \ldots \xrightarrow{\alpha} m_n = m$ in $NP$ going backwards as follows. We start with $\rho = \mu$ (a zero-step run). Let initially $s = s_n$.

Consider the step $m_{n-1} \xrightarrow{\alpha} m_n$ in $NP$. If $s = \{t\}$ is an element autonomous step in $\alpha$, s.t. $\mu' \xrightarrow{t} \mu$, then construct the one-step run $\rho = \mu' \xrightarrow{t} \mu$ in $N$. If $s = (t_0, t_1, \ldots, t_q)$ is a synchronization step, s.t. in $SN$ an arc expression $\gamma(t_0, p)$ contains a variable $x$, $\nu'(x) = \alpha$ and $t = t_j$ $(1 \leq j \leq q)$ is a transition in $\alpha$, where $\mu' \xrightarrow{t} \mu$ (cf. the definition of synchronization step firing), then let $\rho = \mu' \xrightarrow{t} \mu$. If $s = (t_0, t_1, \ldots, t_q)$ is the synchronization step, $\gamma(t_0, p)$ contains a constant $c$, and $\mathcal{I}(c) = \alpha$, then we terminate the construction of $\rho$. For all other cases we skip this step.
Next we similarly consider the previous steps \( m_{n-2} \xrightarrow{s} m_{n-1} \) with \( s = s_{n-1} \) and further in \( NP \), halting after \( s = s_1 \) or meeting a constant. Let \( \rho = \mu' \rightarrow \ldots \mu \) be a run in \( N \) constructed as described above. It is easy to check by induction, that \( (N,\mu') \rightarrow \ldots \mu \rightarrow N \) is either a net token in \( m \), or is a constant in \( NP \), and \( \mu \in \mathcal{R}(N,\mu') \). Since all net tokens in \( m \) and all net constants in \( SN \) are \( \kappa \)-bounded, we get:

\[
\forall p \in P_N : |\mu(p)| \leq \kappa.
\]

So, a place bound for a NP-net \( NP \) with bounded components will be the maximum of place bounds for all components of \( NP \).

Note, that the converse of the Theorem 4.1 is not valid. As a counterexample consider a NP-net with an unbounded net token, which is dead because it cannot synchronize with a system net in any reachable marking.

Theorem 4.1 allows to apply “divide-and-conquer” strategy and is helpful, when components of a NP-net are amenable for checking boundedness because of their size or structural properties.

For example it is known that, if a marked PN has a positive place-invariant, then it is structurally bounded [4]. So, if we succeed in finding positive place invariants for all NP-net components, we can deduce boundedness of the NP-net.

5. Compositionality of liveness for NP-nets with a single synchronization label

Liveness can be defined in several ways for plain Petri nets [15]. We will use the standard “L4-live” variant, which states that every transition in a PN is potentially firable in any reachable marking.

**Definition 5.1.** Let \( t \) be a transition in a component of a marked NP-net \( NP \) with an initial marking \( m_0 \). A transition \( t \) is called live iff for every reachable marking \( m \) there exists a sequence of firings starting from \( m \), which includes \( t \), i.e. \( \forall m \in \mathcal{R}(m_0) : \exists \sigma \in T^* : m \xrightarrow{\sigma} m' \xrightarrow{t} m'' \).

In NP-nets we can distinguish system liveness and liveness of certain agents. Hence we consider two types of liveness. 0L-liveness characterize transitions in the system net.

**Definition 5.2.** Let \( NP \) be a marked two-level NP-net. \( NP \) is called 0L-live iff every transition \( t \) in its system net is live, i.e. \( \forall m \in \mathcal{R}(NP) : \exists \sigma \in (\text{Steps}(NP))^* \exists s \in \text{Steps}(NP) : m \xrightarrow{\sigma} m' \xrightarrow{s} m'' \) and \( t \in s \).

0L-liveness is important when we are more interested in analyzing the system dynamics, rather than behavior of certain agents.

Checking liveness for NP-nets is a complicate problem. Transition firings in different NP-net components are controlled by synchronization mechanism. In addition net-tokens may be consumed by system net transitions. The following theorem gives sufficient conditions for compositionality of 0L-liveness.

**Theorem 5.1.** Let \( NP = (SN, N_1, \ldots, N_k) \) be a marked two-level NP-net with an initial marking \( m_0 \). Let also \( NP \) satisfy the following conditions:

1. \( (SN, m_0) \) is live;

2. all net tokens in \( m_0 \) and all net constants in every arc expression in \( NP \) (considered as separate components) are live;
3. \( NP \) has at most one synchronization label: \(|Lab| \leq 1\).

Then \((NP, m_0)\) is 0L-live.

**Proof:**

Let \( m \) be a reachable marking in \( NP \). Then \( \hat{m} \) is reachable in \( \widehat{SN} \) with the initial marking \( \hat{m}_0 \). Let \( t \) be a transition in \( SN \). Since \( \widehat{SN} \) is live, there is a sequence of firings \( \hat{m} \xrightarrow{t_1} \ldots \xrightarrow{t_n} \) in \( \widehat{SN} \), where \( t_n = t \).

We are to show, that there exists a sequence of firings \( m \xrightarrow{\sigma} \ldots \xrightarrow{\sigma_k} \) in \( NP \), where \( t = s_k^0 \) is the projection of \( s_k \) to transitions in \( SN \). We do it by proving, that for each step \( \hat{m}' \xrightarrow{t'} \hat{m}'' \) in \( \widehat{SN} \) there is a sequence of steps \( m' \xrightarrow{\sigma} m'' \) in \( NP \), s.t. \( \sigma \) includes \( t' \).

This is clearly true, when \( t' \) is labeled with \( \tau \) in \( SN \). If \( t' \) is labeled with \( \lambda \neq \tau \), then it should be a synchronization step, and according to the firing rule for a synchronization step \( t' \) should fire simultaneously with synchronization transitions also labeled with \( \lambda \) in net tokens involved in this firing. Liveness of net tokens then guarantees, that each net token can reach a marking with the enabled transition labeled with \( \lambda \) needed for the synchronization via some sequence of autonomous steps. So, all conditions, needed for a synchronization step can be reached, and a transition \( t' \) labeled with \( \lambda \) can fire. \( \square \)

The second type of liveness asserts liveness of all transitions including transitions in net tokens. It may be helpful, when we are interested in the individual agents behavior as well. A seaport system with ships-tokens can be a good example. This kind of liveness is reasonable, provided net tokens do not disappear during a run execution.

**Definition 5.3.** Let \( NP \) be a NP-net. \( NP \) is called conservative iff for each transition \( t \) in the system net of \( NP \) the set of all variables in the input arc expressions of \( t \) is a subset of all variables in its output arc expressions.

Thus, in conservative NP-nets net tokens can change their markings, but they cannot disappear.

**Definition 5.4.** Let \( NP \) be a marked conservative NP-net. \( NP \) is said to be 1L-live iff every transition \( t \) in \( NP \) (either in the system net of \( NP \), or in one of the net tokens from the initial marking in \( NP \), or in a net constant in one of arc expressions in \( NP \)) is live, i.e. \( \forall m \in R(NP) : \exists \sigma \in (Steps(NP))^* \exists s \in Steps(NP) : m \xrightarrow{\sigma} m' \xrightarrow{s} \) and \( t \in s \).

The following theorem gives sufficient conditions for compositionality of 1L-liveness.

**Theorem 5.2.** Let \( NP = (SN, N_1, \ldots, N_k) \) be a marked NP-net with an initial marking \( m_0 \). Let also \( NP \) satisfy the conditions 1-3 from the Theorem 5.1, as well as the following condition:

if an element net \( N = N_i \) contains a transition labeled with \( \lambda \neq \tau \), then for each place \( p \) typed with \( N \) in the system net \( SN \) of \( NP \) there is a path \( \pi \) from \( p \) to some transition \( t \) labeled with \( \lambda \), such that

1. all places in \( \pi \) are typed with \( N \);

2. for each transition \( t' \) on \( \pi \) the expressions on two arcs leading to/from \( t' \) upon \( \pi \) contain at least one common variable.

Then \( NP \) is 1L-live.
Proof:
Liveness of transitions in the system net follows from the Theorem 5.1. Additionally we should prove liveness of transitions in net tokens.

Let \( m \in \mathcal{R}(NP) \), \( p \in P_{SN} \), and let \( t \) be a transition in a net token \( \alpha = (N, \mu) \), where \( \alpha \in m(p) \).

Since in a system net each variable in an output arc expression for a transition should occur in one of its input arc expressions, there is an exact constant \( c \) in an arc expression in \( NP \), s.t. \( I(c) = (N, \mu_0) \) and \( \mu \in \mathcal{R}(\hat{N}, \mu_0) \), or \( m_0 \) contains a net token \( (\hat{N}, \mu_0) \), s.t. \( \mu \in \mathcal{R}(\hat{N}, \mu_0) \). Then \( (\hat{N}, \mu) \) is live, and there is a sequence of firings \( \mu \xrightarrow{t_1} \ldots \xrightarrow{t_k} \) in \( \hat{N} \), where \( t_k = t \).

We prove now, that for each step \( \mu \xrightarrow{t_i} \mu_{i+1} \) from \( \mu \xrightarrow{t_1} \ldots \xrightarrow{t_k} \) there is a sequence of steps \( m_i \xrightarrow{\sigma_i} m_{i+1} \) in \( NP \), s.t. \( \sigma_i \) includes \( t_i \) and \( (N, \mu_i), (N, \mu_{i+1}) \) are net tokens in some reachable in \( NP \) markings \( m_i, m_{i+1} \) correspondingly.

The case when \( t_i \) is labeled with \( \tau \) and thus corresponds to an element autonomous step, is trivial. Let \( t_i \) be labeled with \( \lambda \neq \tau \). Let the net token \( \alpha = (N, \mu_i) \) reside in a place \( p \) in \( m_i \). Under the hypothesis of the theorem there is a path \( \pi \) from \( p \) to some \( t^\lambda \in T_{SN} \), labeled with \( \lambda \) in \( SN \). The system net \( SN \) is live, so all transitions on \( \pi \) can be sequentially fired (involving \( N \) in each firing) possibly alternately with some additional transitions. The transition \( t_i \) in \( N \) will be fired in these steps either in the very last step, or earlier. We break constructing \( \sigma_i \) as soon as \( t_i \) has fired. \( \square \)

6. Compositionality of liveness for NP-nets with several synchronization labels

The case of NP-nets with more then one synchronization label turns to be more complex. The problem with establishing liveness of NP-nets in this case is connected with possible in-coordination of net tokens behavior and system net behavior. To deal with this problem we introduce the notion of must-bisimilarity (m-bisimilarity for short), which may be considered as a loose variation of weak bisimilarity.

Let \( N = (S, T, F) \) be a Petri net with transitions labeled by labels from \( Lab \cup \{ \tau \} \), where \( \tau \) is a special label for a silent action. By abuse of notation we write \( m \xrightarrow{\lambda} m' \), if firing of a transition \( t \), labeled with \( \lambda \in Lab \cup \{ \tau \} \), transfers a marking \( m \) to a marking \( m' \). For \( \lambda \neq \tau \) we write \( m \xrightarrow{\lambda} m' \) if there is a sequence of firings \( m = m_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} m_k \xrightarrow{\lambda} m^1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} m^n = m' \), where \( k, n \geq 1 \). We write \( m \xrightarrow{\tau} m' \) if there is a sequence of firings \( m = m_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} m_k = m' \), where \( k \geq 1 \).

Definition 6.1. Let \( N_1 = (S_1, T_1, F_1, m_0^1) \) and \( N_2 = (S_2, T_2, F_2, m_0^2) \) be two marked Petri nets with transitions labeled by alphabet \( Lab \cup \{ \tau \} \). Nets \( N_1 \) and \( N_2 \) are m-bisimilar (denoted \( N_1 \approx_m N_2 \)) iff there is a relation \( R \subseteq \mathcal{R}(N_1) \times \mathcal{R}(N_2) \) such that

1) \((m_0^1, m_0^2) \in R;\)
2) for any \((m_1, m_2) \in R\), if \( m_1 \xrightarrow{\lambda} m_1' \) in \( N_1 \), then
   (a) there exists \( m_2' \) such that \( m_2 \xrightarrow{\lambda} m_2' \) in \( N_2 \) and \((m_1', m_2') \in R;\)
   (b) for each \( m_2' \) such that \( m_2 \xrightarrow{\lambda} m_2' \) in \( N_2 \) it should be \((m_1', m_2') \in R;\)
3) and symmetrically, for any \((m_1, m_2) \in R\), if \( m_2 \xrightarrow{\lambda} m_2' \) in \( N_2 \), then
We start building the call such a net an \( \alpha \)-trail net with a new place \( p' \), corresponding to \( p_\alpha \), and consider \( p_\alpha \) as a current place in \( SN \).

(a) there exists \( m'_1 \) such that \( m_1 \xym{k} m'_1 \) in \( N_1 \) and \( (m'_1, m'_2) \in R \),

(b) for each \( m'_1 \) such that \( m_1 \xym{k} m'_1 \) in \( N_1 \) it should be \( (m'_1, m'_2) \in R \).

To ensure 1L-liveness of a NP-net with more than one synchronization label we require \( R \)-bisimilarity between a net token \( \alpha \) and a colored Petri net, describing possible trajectories of \( \alpha \). We call such a net an \( \alpha \)-trail net.

Now we describe an **algorithm for constructing an \( \alpha \)-trail net** for a given marked system net with a net token \( \alpha \), residing in a place \( p \) of the system net.

Let \( NP \) be a conservative NP-net with a system net \( SN \). Let \( p_\alpha \) be a place in the system net, where a net token \( \alpha \) is located in the initial marking. By abuse of notation by \( \text{Var}(e) \) we denote the set of all variables in the expression \( e \in Expr \).

The algorithm, represented in the table, informally can be described as follows.

**Step 0.** We start building the \( \alpha \)-trail net with a new place \( p' \), corresponding to \( p_\alpha \), and consider \( p_\alpha \) as a current place in \( SN \).
Step 1. Let \( p \) be a current place, and let \( t \) be a transition in \( SN \), for which \( p \) is an input place, i.e. there is an arc from \( p \) to \( t \) with an arc expression \( expr \).

Let \( V \) be the set of variables occurring in \( expr \). For each \( v \in V \), such that there is an outgoing arc from \( t \) with an expression containing \( v \), we build a new transition \( t' \) and a new place \( p' \):

\[
\begin{align*}
& t' = p, \\
& t'^* = p' \text{ and } \lambda(t') = \lambda(t) \quad (\text{Cf. an example in Fig. 3}).
\end{align*}
\]

Step 2. Repeat Step 1 with every new place as a current place.

![Figure 3](image-url) A picture example for the Step 2 of the algorithm.

Fig. 4 shows an example of building an \( \alpha \)-trail net for a NP-net with the system net \( SN \) and \( \alpha \), residing in \( p_1 \).

**Theorem 6.1.** Let \( NP \) be a conservative marked NP-net with a system net \( SN \) and an initial marking \( m_0 \). Let also \( NP \) satisfy the following conditions:

1. \((\hat{SN}, \hat{m}_0)\) is live;
2. all net tokens in \( m_0 \) and all net constants in every arc expression in \( NP \) (considered as separate components) are live;
3. for each net token \( \alpha \) in \( m_0 \), residing in a place \( p \): \( \alpha \) is \( m \)-bisimilar to the \( \alpha \)-trail net for \( p \);
4. for each arc \((t, p)\) with an arc expression \( e \) in \( SN \): if \( e \) contains a net constant with a value \( \alpha \), then \( \alpha \) is \( m \)-bisimilar to the \( \alpha \)-trail net for \( p \).

Then \( NP \) is 1L-live.

**Proof sketch:**
Let \( m \) be a reachable marking in \( NP \). It can be checked by induction, that all net tokens in \( m \) are live, and for each net token \( \alpha \) residing in a place \( p \) in \( m \), \( \alpha \) is \( m \)-bisimilar to the \( \alpha \)-trail net for \( p \).

Let \( t \) be a transition in \( SN \). Since \( \hat{SN} \) is live, there is a sequence of firings \( \tilde{m} \xrightarrow{t_1} \cdots \xrightarrow{t_n} \hat{m} \) in \( \hat{SN} \), where \( t_n = t \). We show, that for each step \( \tilde{m}' \xrightarrow{t'} \tilde{m}'' \) in \( \hat{SN} \) from this sequence, there is a sequence of steps \( m' \xrightarrow{\sigma} m'' \) in \( NP \), s.t. \( \sigma \) includes \( t' \). If \( t' \) is labeled with \( \tau \), then \( \sigma = \{t'\} \) consists of one
system autonomous $t'$-firing. If $t'$ is labeled with $\lambda \neq \tau$, then $t'$ is enabled in $\widehat{SN}$ and all net tokens in its pre-places are $m$-bisimilar to their $\alpha$-trails. It means, that (possibly after several firings of element autonomous steps) each of them can fire a transition labeled with $\lambda$.

The case, when $t$ is a transition in a net token, is proved in the similar way.

7. Conclusion

In this paper we have characterized conditions under which boundedness and liveness for nested Petri nets can be deduced in a compositional way, i.e. from boundedness, respectively liveness, of nested net components. Compositional approach allows to reduce crucially a state space, and makes checking these properties a tractable problem for nested Petri nets with relatively simple components.

These results may be helpful for analysis of NP-nets with components small enough for model checking, or components from some restricted Petri net subclasses. For example for free-choice Petri nets boundedness and liveness can be checked in polynomial time. In Section 2 we presented a NP-net model of a P2P protocol with free-choice components, for which boundedness and liveness can be easily established.

The further research will concern looking for other behavioral properties of nested Petri nets, which can be deduced in a compositional way.

The authors would like to thank the anonymous referees for very helpful remarks.
References


