Algorithm for Computing Unfoldings of Unbounded Hybrid Petri Nets

Petr Novosad and Milan Češka

Faculty of Information Technology, Brno University of Technology,
Božetěchova 2, 612 66 Brno, Czech Republic
{novosad,ceska}@fit.vutbr.cz

Abstract. The paper formalizes the concept of the unfolding for unbounded hybrid Petri nets and introduces the algorithm for its computing. The unfolding is a useful partial-order based method for analysis and verification of the Petri net properties. This technique can cope well with the so-called state space explosion problem, especially for the Petri nets with a lot of concurrency.

Keywords: Hybrid Petri net, unfolding.

1 Introduction

Petri nets are a mathematical and graphical tool for modeling concurrent, parallel and/or distributed systems. An unfolding is a useful structure for checking properties of the Petri nets. Our goal it to update the algorithm for computing unfolding for discrete Petri nets to continuous and unbounded hybrid Petri nets.

This article extends our previous work [14] by introducing unfoldings for ordinary unbounded hybrid Petri nets. Unbounded hybrid Petri nets have infinite state space and thus similar set of problems with reachability arises as for unbounded discrete Petri nets [12].

The article consists of the following. The definitions and notations of the hybrid Petri nets are given in Section 2. Section 3 contains definitions and notations of unfoldings. Section 4 presents algorithm for unfolding construction with examples. Section 5 concludes the paper.

2 Hybrid Petri Nets

The concept of the continuous and hybrid Petri nets has been presented by David and Alla in 1987 [3,5,6,4]. It is a fluidification of the discrete Petri net. Some places can hold a real valued marking. This paper assumes that the reader is familiar with the basic theory of the Petri nets [1,2]. The Petri net is *persistent* when enabled transitions can only be disabled by its own firing.

2.1 Continuous Petri Nets

Continuous Petri net [6] is defined as a 5-tuple $R_C = (P, T, Pre, Post, M_0)$, where P is a finite set of places and T is a finite set of transitions. $P \neq \emptyset$,

R. Moreno-Díaz et al. (Eds.): EUROCAST 2013, Part I, LNCS 8111, pp. 428–435, 2013.

$T \neq \emptyset$ and $P \cap T = \emptyset$. $Pre : P \times T \to \mathbb{Q}^+$ is the input incidence matrix. $Post : P \times T \to \mathbb{Q}^+$ is the output incidence matrix. $M_0 : P \to \mathbb{R}^+$ is the initial marking[1]. Let $p \in P, t \in T$: $Pre(p,t)$ is the weight of the arc $p \to t$; $Post(p,t)$ is the weight of the arc $t \to p$. If the arc does not exist, the weight is 0. In a graphical representation of the continuous Petri net places are represented by double circles and transitions are represented by empty rectangles (Fig. 1).

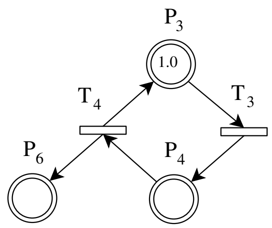

Fig. 1. The unbounded continuous Petri net

The *continuous marking* $m \in (\mathbb{R}^+)^{|P|}$ is a vector of non-negative real numbers. A transition $t \in T$ is *enabled* in a marking m, iff $\forall p \in {}^\bullet t : m(p) > 0$. Enabling of the transition does not depend on the arc weight, it is sufficient that every input place has a non-zero marking. The *enabling degree* q of the transition t for the marking m is the maximal amount that the transition can fire in one go, i.e. $q(t,m) = \min_{p \in {}^\bullet t} (m(p)/Pre(p,t))$. Firing the transition t with a quantity $\alpha < q(t,m), \alpha \in \mathbb{R}^+$ is denoted as $m \overset{\alpha t}{\to} m'$. $[t]^\alpha$ represents $\alpha \in \mathbb{R}^+$ firings of the transition t at one go. The new marking $m' = m + \alpha.C(P,t)$, where $C = Post - Pre$ is a token-flow matrix. The marking m' is *reachable* from the marking m.

Let m be a marking. The set P of places may be divided into two subsets: $P^+(m)$ the set of places $p \in P$ such that $m(p) > 0$, and the set of places p such that $m(p) = 0$. A *continuous macro-marking* is the union of all markings m with the same set $P^+(m)$ of marked places. Since each continuous macro-marking is based on the Boolean state of every place (marked or not marked), the number of continuous macro-markings is less than or equal to 2^n, where n is the number of places.

2.2 Hybrid Petri Nets

Hybrid Petri net [6] is a 6-tuple $R_H = (P, T, Pre, Post, M_0, h)$, where P is a finite set of discrete and continuous places, T is a finite set of discrete and continuous transitions. $P \neq \emptyset$, $T \neq \emptyset$ and $P \cap T = \emptyset$. $Pre : P \times T \to \mathbb{Q}^+$ or $\mathbb{N}$ is the input incidence matrix. $Post : P \times T \to \mathbb{Q}^+$ or $\mathbb{N}$ is the output incidence matrix.

[1] Notation $\mathbb{Q}^+$ corresponds to the non-negative rational numbers and notation $\mathbb{R}^+$ corresponds to the non-negative real numbers (both including zero).

Let $p \in P, t \in T$: $Pre(p,t)$ is the weight of the arc $p \to t$; $Post(p,t)$ is the weight of the arc $t \to p$. If the arc does not exist, the weight is 0. A graphical representation of the hybrid Petri net is shown in Fig. 2. $M_0 : P \to \mathbb{R}^+$ or $\mathbb{N}$ is the *initial marking.* A function $h : P \cup T \to \{D, C\}$ is called a *hybrid function,* that indicates for every node whether it is a discrete node (sets P_D and T_D) or a continuous one (sets P_C and T_C). In the definitions of $Pre, Post$ and m_0, the set $\mathbb{N}$ corresponds to the case where $p \in P_D$ and the set $\mathbb{Q}^+$ to the case where $p \in P_C$. For the discrete places $p \in P_D$ and the continuous transitions $t \in T_C$ must hold $Pre(p,t) = Post(p,t)$.

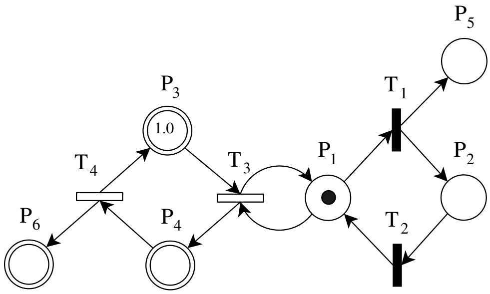

Fig. 2. The unbounded hybrid Petri net

The *hybrid marking* for the hybrid Petri net is a couple $m = (m_D, m_C)$, where m_D denotes the marking of the discrete places and m_C denotes the continuous macro-marking of the continuous places. The discrete transition $t \in T_D$ is *enabled* in a marking m, iff $\forall p \in {}^{\bullet}t : m(p) \geq Pre(p,t)$. The *enabling degree* q of the discrete transition t for the marking m is integer $q(t,m) = \min_{p \in {}^{\bullet}t} \; (m(p)/Pre(p,t))$. For continuous places $p \in {}^{\bullet}t \wedge p \in P_C$ the edge $p \to t$ is a treshold for marking in the place p for enabling the discrete transition t. A continuous transition $t \in T_C$ is *enabled* in a marking m, iff $\forall p \in {}^{\bullet}t \wedge p \in P_D$: $m(p) \geq Pre(p,t)$ and $\forall p \in {}^{\bullet}t \wedge p \in P_C : m(p) > 0$. The *enabling degree* q of the continuous transition t for the marking m is $q(t,m) = \min_{p \in {}^{\bullet}t} \; (m(p)/Pre(p,t))$.

3 Unfoldings

The unfolding [9,7,8,10,11] is a useful partial-order method for analysis and verification of the Petri net properties. This technique can cope well with the so-called state space explosion [12], specially for the Petri nets with a lot of concurrency. The state space of the Petri net is represented by an acyclic net with a simpler structure than the Petri net. The unfolding represents all reachable states of the Petri net and can be infinite if the Petri net has a cycle. However it can be truncated before it starts to repeat.

Our approach combines the macro-markings from the so-called case graph for the continuous Petri nets [6,13] with the idea of the coverability unfolding for the unbounded discrete Petri nets [15]. Continuous conditions in the unfolding can have associated a symbol representing the macro-marking thus some nonzero real marking. Discrete conditions in the unfolding can have associated a symbol ω representing that the corresponding place is unbounded.

A *net* is a triple $N = (P, T, F)$, where P is a finite set of places and T is a finite set of transitions. $P \neq \emptyset$, $T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation.

An *occurrence net* is a net $O = (B, E, G)$, where B is a set of occurence of places, E is a set of occurrence of transitions. O is acyclic and G is the acyclic flow relation, i.e. for every $x, y \in B \cup E : xG^+y \Rightarrow \neg yG^+x$, where G^+ is a transitive closure of G. Let us denote $x < y$, iff xG^+y, and $x \leq y$, iff $x < y$ or $x = y$. The relation $<$, resp. $\leq$ is a partial order relation. Nodes $x, y \in (P \cup T)$ are in a *conflict* relation, denoted by $x\#y$, iff $\exists t_1, t_2 \in T : t_1 \neq t_2 \wedge^\bullet t_1 \cap^\bullet t_2 \neq \emptyset \wedge t_1 \leq x \wedge t_2 \leq y$. Nodes $x, y \in (P \cup T)$ are in a *concurrency* relation, denoted by $x\ co\ y$, if neither $x < y$ nor $y < x$ nor $x\#y$. For every $b \in B : |^\bullet b| \leq 1$. For every $x \in (B \cup E) : \neg(x\#x)$, i.e. no element is in confict with itself. The set of elements $\{y \in (B \cup E)|y < x\}$ is finite, i.e. O is finitely preceded. $Min(O)$ denotes the set of minimal elements of $B \cup E$ with respect to the relation $\leq$, i.e. the elements with an empty preset.

A *homomorfism* from the occurrence net O to the hybrid Petri net $R_H = (P, T, Pre, Post, M_0, h)$ is a mapping $p : B \cup E \rightarrow P \cup T$ such that $p(B) \subseteq P$ and $p(E) \subseteq T$, i.e. preserves the nature of nodes. For every $e \in E : p(^\bullet e) =^\bullet p(e) \wedge p(e^\bullet) = p(e)^\bullet$, i.e. p preserves the environment of transitions. The restriction of p to $Min(O)$ is a bijection between $Min(O)$ and M_0.

A *hybrid branching process* of the unbounded hybrid Petri net R_H is a quadruple $\pi_H = (B, E, G, p, d, w) = (O, p, d, w)$, where O is the labelled occurrence net and $p(x) = y$ denotes labelling element x as element y. A mapping $d : E \rightarrow \{m_1, \ldots, m_{|P|}\} \cup \{0\}$ labels transitions occurrences with symbol m_i indicating maximal firing degree or with 0 indicating arbitrary lower degree (that will not be depicted). A mapping $w : B \rightarrow \{\omega, 1\}$ labels discrete places occurrences with symbol ω indicating an unbounded discrete place or with 1 otherwise (that will not be depicted). The type of the node determines its graphical representation. Every node $e \in E : p(e) \in T_C$ is represented by double rectangle and every node $b \in B : p(b) \in P_C$ is represented by double circle with the name of the corresponding marking.

A hybrid branching process $\pi'_H = (O', p', d', w')$ is a *prefix* of π_H, denoted by $\pi'_H \sqsubseteq \pi_H$, if $O' = (B', E', G')$ is a subnet of O satisfying $Min(O)$ belongs to O'; if $e \in E'$ and $(b, e) \in G$ or $(e, b) \in G$ then $b \in B'$; if $b \in B'$ and $(e, b) \in G$ then $e \in E'$; p' is the restriction of p to $B' \cup E'$. For every R_H there exists a unique (up to isomorphism) maximal (w.r.t. $\sqsubseteq$) branching process that is called *unfolding.*

A *configuration* of the occurrence net O is a set of the transitions occurrences $C \subseteq E$ such that for all $e_1, e_2 \in C : \neg(e_1\#e_2)$, i.e. C is conflict-free. For every $e_1 \in C : e_2 \leq e_1 \Rightarrow e_2 \in C$, i.e. C is causally closed. A *local configuration* $[e]$ for the transition occurrence $e \in E$ is a set $[e] = \{e' \in E|e' \leq e\}$.

A set of places occurrences $D \subseteq B$ is called a *co-set*, iff for all distinct $d_1, d_2 \in D : d_1 \; co \; d_2$. A *cut* is the maximal (w.r.t. set inclusion) co-set. For every $d_1, d_2 \in D$, if $p(d_1) = p(d_2)$ then $d_1 = d_2$. Let C be the finite configuration of the hybrid branching process π_H. Then $Cut(C) = (Min(O) \cup C^\bullet) \setminus {}^\bullet C$ is a cut. A set $Mark(C) = p(Cut(C))$ is the reachable hybrid macro marking of the hybrid Petri net R_H.

An *adequate order* $\triangleleft$ is a strict well-founded partial order on the local configurations such that for two transitions occurrences $e_1, e_2 \in E : [e_1] \subset [e_2] \Rightarrow [e_1] \triangleleft [e_2]$. The transition occurrence $e_1 \in E$ is a *cut-off* transition induced by $\triangleleft$, iff there is a corresponding transition $e_2 \in E$ with $Mark([e_1]) = Mark([e_2])$ and $[e_2] \triangleleft [e_1]$. The order $\triangleleft$ is a refined partial order from [9]. For the hybrid branching process π_H and every $e_1, e_2 \in E : p(e_1) \in T_D \wedge p(e_2) \in T_C \Rightarrow [e_1] \triangleleft [e_2]$. For every $e_1, e_2 \in E : d(e_1) \neq 0 \wedge d(e_2) = 0 \Rightarrow [e_1] \triangleleft [e_2]$.

The hybrid branching process is *complete*, iff for every reachable hybrid macro marking $M \in [M_0 >$ of the hybrid Petri net R_H there is the configuration C of π_H such that $M = Mark(C)$ and for every transition $t \in T$ enabled in M there is the finite configuration C and the transition occurrence $e \in C$ such that $M = Mark(C)$, $p(e) = t$ and $C \cup \{e\}$ is the configuration.

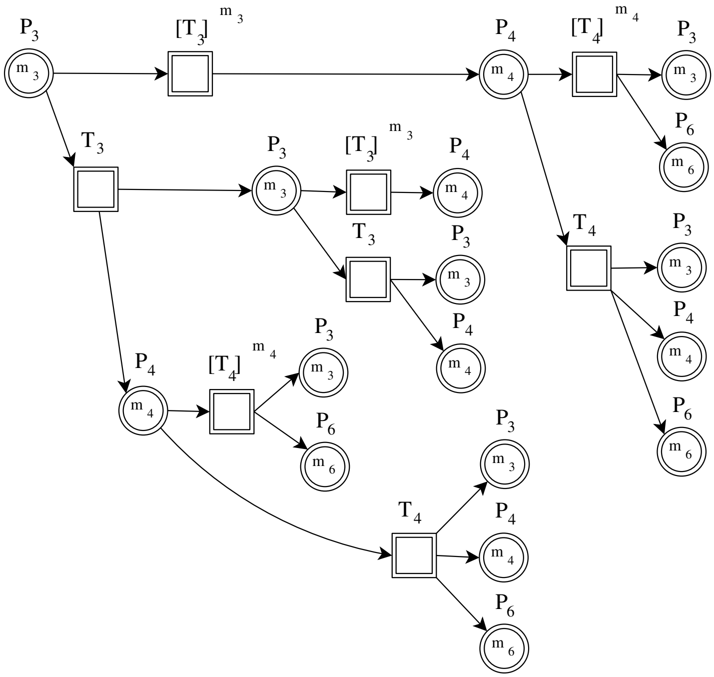

Fig. 3. The prefix of the unfolding of the unbounded continuous Petri net from Fig. 1

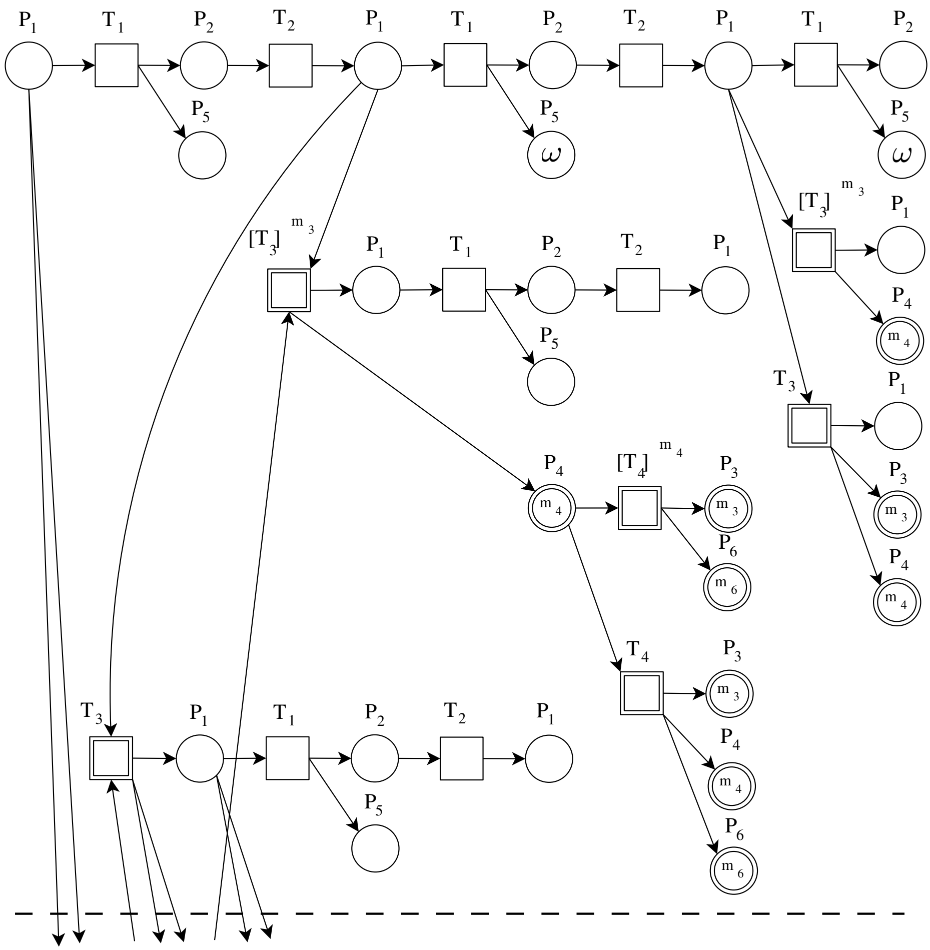

Fig. 4. The main segment of the finite prefix of the unfolding of the unbounded hybrid Petri net from Fig. 2. The whole prefix is not depicted because of size limitations.

4 Algorithm

The algorithm 1 is a modified and extended algorithm presented in [8]. It constructs the finite and complete prefix of the unfolding of the unbounded hybrid Petri net. A function *InitializePrefix()* initializes the prefix *pref* with instances of the places from M_0. A function *PossibleExtensions()* finds the set of possible extensions of the hybrid branching process *pref* using possible transitions firings for the hybrid Petri net, including continuous transitions firings with the maximal degree. The decision version of this function is NP-complete in the size of the prefix *pref*. A function *MinimalExtension()* chooses the transition occurrence with minimal local configuration with respect to the order $\triangleleft$ from the set

Algorithm 1. The finite prefix of the unfolding for the unbounded hybrid Petri net.

```
Input: The unbounded hybrid Petri net R_H = (P, T, Pre, Post, M_0, h)
Output: The finite prefix pref = (O, p, d, w) of the unfolding
begin
    InitializePrefix(pref);
    pe = PossibleExtensions(pref);
    cutoff = ∅;
    while pe ≠ ∅ do
        e = MinimalExtension(pe);
        if [e ] ∩ cutoff = ∅ then
            Extend(pref, e);
            pe = PossibleExtensions(pref);
            if IsCutoff(e) then cutoff = cutoff ∪ {e };
        else
            pe = pe \ {e };
        end
    end
end
```

of possible extensions. A function *Extend()* appends new instance of the transition occurrence and new instances of the output places of the transition. The function also detects an unbounded discrete place by comparing the new and the previous discrete state. The label of the unbounded discrete place is propagated further once denoted. A function *IsCutoff* determines whether the transition occurrence is a cut-off transition. The algorithm is finite because the number of continuous, resp. discrete macro markings in the continuous, resp. discrete part of the hybrid Petri net is finite and it transforms all transitions occurrences into cut-off transitions [9].

An example of the finite prefix of the unfolding created by the algorithm 1 for the unbounded continuous Petri net in Fig. 1 is in Fig. 3. All reachable continuous macro markings are represented by cuts.

The image in Fig. 2 shows very simple, yet typical example from the application domain of the hybrid Petri nets, where the discrete part enables or disables the continuous transitions. An example of the complete and finite prefix of the unfolding created by the algorithm 1 for the unbounded hybrid Petri net in Fig. 2 is in Fig. 4. The image shows only the most interesting part of the whole prefix because of size limitations. It can be seen how the unbounded discrete place is detected and propagated further.

5 Conclusion and Future Work

We have introduced the algorithm for computation of the unfolding for the ordinary unbounded hybrid Petri nets and shown the corresponding definitions. Some information regarding reachability is lost due to the abstraction in the continuous and discrete macro markings. Nevertheless, advantages of the unfolding remain. Analysis of the partial order between the transitions occurrences and checking on persistency by analysing the conflicts between the transitions occurrences in the unfolding is simpler due to absence of cycles. It preserves concurrency and explicitly represents conflicts.

In the future we plan to develop algorithms for analysing properties of the hybrid Petri nets from the unfolding.

Acknowledgements. This work was supported by the Czech Science Foundation (P103/10/0306), the Czech Ministry of Education (MSM 0021630528) and the EU/Czech IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070.

References

1. Reisig, W.: Petri Nets - an Introduction, Berlin (1985)
2. Desel, J., Juhás, G.: What Is a Petri Net? In: Ehrig, H., Juhás, G., Padberg, J., Rozenberg, G. (eds.) APN 2001. LNCS, vol. 2128, pp. 1–25. Springer, Heidelberg (2001)
3. David, R., Alla, H.: Continuous Petri nets. In: Proc. of the 8th European Workshop on Application an Theory of Petri Nets, Zaragoza, Spain, pp. 275–294 (1987)
4. Recalde, L., Teruel, E., Silva, M.: Autonomous Continuous P/T Systems. In: Donatelli, S., Kleijn, J. (eds.) ICATPN 1999. LNCS, vol. 1639, pp. 107–126. Springer, Heidelberg (1999)
5. David, R., Alla, H.: Continuous and hybrid Petri nets. Journal of Circuits, Systems, and Computers (1998)
6. David, R., Alla, H.: Discreet, Continuous and hybrid Petri nets. Springer, Berlin (2005)
7. McMillan, K.L.: A Technique of State Space Search Based on Unfolding. Formal Methods in System Design 6, 45–65 (1995)
8. Esparza, J., Romer, S., Vogler, W.: An Improvement of McMillan's Unfolding Algorithm. Formal Methods in System Design 20 (2002)
9. Esparza, J., Heljanko, K.: Unfoldings - a partial-order approach to model checking. EATCS Monographs in Theoretical Computer Science. Springer (2008)
10. Khomenko, V., Koutny, M.: Towards an Efficient Algorithm for Unfolding Petri Nets. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 366–380. Springer, Heidelberg (2001)
11. Weidlich, M., Elliger, F., Weske, M.: Generalised Computation of Behavioural Profiles Based on Petri-Net Unfoldings. In: Bravetti, M., Bultan, T. (eds.) WS-FM 2010. LNCS, vol. 6551, pp. 101–115. Springer, Heidelberg (2011)
12. Valmari, A.: The State Explosion Problem. In: Reisig, W., Rozenberg, G. (eds.) APN 1998. LNCS, vol. 1491, pp. 429–528. Springer, Heidelberg (1998)
13. Novosad, P., Češka, M.: Algorithms for Computing Coverability Graphs for Hybrid Petri Nets. In: MEMICS, pp. 177–183. MUNI, Brno (2008)
14. Novosad, P., Češka, M.: Unfoldings of Bounded Hybrid Petri Nets. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2011, Part I. LNCS, vol. 6927, pp. 543–550. Springer, Heidelberg (2012)
15. Desel, J., Juhás, G., Neumair, C.: Finite Unfoldings of Unbounded Petri Nets. In: Cortadella, J., Reisig, W. (eds.) ICATPN 2004. LNCS, vol. 3099, pp. 157–176. Springer, Heidelberg (2004)