Newcastle
University

COMPUTING
SCIENCE

Coverability and Inhibitor Arcs: an example

Sander van der Vlugt, Jetty Kleijn and Maciej Koutny

TECHNICAL REPORT SERIES

No. CS-TR-1293 November 2011

TECHNICAL REPORT SERIES

No. CS-TR-1293 November, 2011

Coverability and Inhibitor Arcs: an example

S. van der Vlugt, J. Kleijn, M. Koutny

Abstract

Inhibitor arcs pose a problem for the standard Coverability Tree Construction for Place/Transition Nets. A straightforward modification of the construction circumventing this problem works for PT-nets with one inhibitor place. Here it is shown that this modified construction may not terminate in case of two or more inhibitor places.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

VAN DER VLUGT, S., KLEIJN, J., KOUTNY, M.
Coverability and Inhibitor Arcs: an example
[By] S. van der Vlugt, J. Kleijn, M. Koutny
Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1293)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1293

Abstract

Inhibitor arcs pose a problem for the standard Coverability Tree Construction for Place/Transition Nets. A straightforward modification of the construction circumventing this problem works for PT-nets with one inhibitor place. Here it is shown that this modified construction may not terminate in case of two or more inhibitor places.

About the authors

Sander van der Vlugt is an MSc student at the Leiden University, The Netherlands.

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny obtained his MSc (1982) and PhD (1984) from the Warsaw University of Technology. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and in 1994 was promoted to an established Readership at Newcastle. In 2000 he became a Professor of Computing Science.

Suggested keywords

PETRI NET
INHIBITOR ARC
COVERABILITY

Coverability and Inhibitor Arcs
an example

Sander van der Vlugt[1], Jetty Kleijn[1] and Maciej Koutny[2]

[1] LIACS, Leiden University
P.O.Box 9512, NL-2300 RA Leiden, The Netherlands
svdvlugt@liacs.nl, kleijn@liacs.nl
[2] School of Computing Science, Newcastle University
Newcastle upon Tyne NE1 7RU, U.K.
maciej.koutny@ncl.ac.uk

Abstract. Inhibitor arcs pose a problem for the standard Coverability Tree Construction for Place/Transition Nets. A straightforward modification of the construction circumventing this problem works for PT-nets with one inhibitor place. Here it is shown that this modified construction may not terminate in case of two or more inhibitor places.

1 Introduction

Coverability trees are a powerful tool in the behavioural analysis of Petri Nets. Originally introduced in [5] for Vector Addition Systems, they proved to be useful also for Place/Transition Nets (PT-nets) [3]. A coverability tree provides a finite representation of the reachable markings of a PT-net from which important behavioural properties of the net can be derived [8, 7], such as coverability of markings, boundedness, and mutual exclusion properties. Inhibitor arcs are a powerful extension of PT-nets, as they allow to check for emptiness of a place (so-called zero-testing), a feature not possible in the standard PT-net model. Actually, Place/Transition Nets with inhibitor arcs (PTI-nets) can simulate Turing machines [1] and so, many problems that are decidable for PT-nets, like reachability and boundedness, are undecidable for PTI-nets [4].

In [6], it is investigated how the standard coverability tree construction for PT-nets might be generalised to PT-nets and PTI-nets working under the a priori step semantics. This has led to a generic algorithm for the construction of so-called *step coverability trees*. Even without the step semantics, inhibitor arcs introduce the problem of non-monotonicity and to deal with this, [6] first proposes a straightforward modification of the standard coverability tree construction.

As part of his BSc project, the first author investigated this modification, aimed at dealing with PT-nets with inhibitor arcs. The algorithm correctly identifies unbounded places, and always terminates in case of one inhibitor place. As shown in [6], there exists however a PT-net with three inhibitor places for which the algorithm does not terminate. It was not known whether the algorithm would always terminate in case of two inhibitor places. This paper solves this problem by giving an example of a PT-net with two inhibitor places for which the algorithm does not terminate.

2 Preliminaries

A *P/T-net with inhibitor arcs* (PTI-net) is a tuple $\mathcal{N} = (P, T, W, I, M_0)$, where P and T are disjoint finite sets of *places* and *transitions*, respectively; $W : (T \times P) \cup (P \times T) \to \mathbb{N}$ is the *weight function*; $I \subseteq P \times T$ is the set of *inhibitor arcs*; and $M_0 : P \to \mathbb{N}$ is the *initial marking* (in general, any mapping $M : P \to \mathbb{N}$ is a marking). In diagrams places are drawn as circles, transitions as rectangles, the weight function is represented by (weighted) arcs, and inhibitor arcs are drawn with a small circle as arrowhead. A marking M is represented by placing $M(p)$ *tokens* (small black dots) inside the circle representing place p.

A transition $t \in T$ is *enabled* at a marking M if $M(t) \geq W(p,t)$, for every $p \in P$, and $M(p) = 0$, for every $p \in I(t) = \{q \mid (q,t) \in I\}$ (i.e., all inhibitor places of t are empty). In such a case t can *fire* leading to the marking M' satisfying $M'(p) = M(p) - W(p,t) + W(t,p)$, for every $p \in P$. We denote this by $M[t\rangle M'$. The enabling condition and firing of a transition can be generalised in the obvious way to *extended* markings defined as mappings $M : P \to \mathbb{N} \cup \{\omega\}$, where ω is the smallest infinite ordinal.

A marking M is *reachable* if it can be obtained from the initial marking through successively firing a finite sequence of transitions, and it is *coverable* if there is a reachable marking M' such that $M'(p) \geq M(p)$, for all $p \in P$.

A *PT-net* net is nothing but a PTI-net without any inhibitor arcs.

3 Constructing Coverability Trees for PTI-nets

The algorithm shown in Table 1, but without the line indicated by ($*$), is basically the standard coverability tree construction introduced in [5]. If the input net N is a PT-net, i.e. $I = \emptyset$, then the line indicated by ($*$) is void. It generates a tree $CT = (V, A, \mu, v_0)$, where V is a set of nodes ($v_0 \in V$ is the root), A is a set of arcs labelled with the transitions of N, and μ is a mapping associating a (possibly extended) marking with each node in V. An arc labelled t from node v to node w is denoted by $v \xrightarrow{t} w$, and $v \rightsquigarrow^{\sigma}_{A} w$ means that node w can be reached from node v with σ being the sequence of transitions labelling the arcs along the path.

The algorithm starts with a single (root) node, corresponding to the initial marking. Then, repeatedly, for each transition that is enabled at a marking corresponding to an already generated but not yet processed node, an arc and a new node representing the resulting marking are added. If the latter already appears in the tree, the new node is ignored and otherwise it becomes an unprocessed node. For a PT-net, place unboundedness is detected whenever a marking corresponding to a new node M strictly *covers* the marking M' corresponding to an ancestor node (i.e., $M' < M$ meaning that $M' \neq M$ and $M'(p) \leq M(p)$ for all places p). In such a case, the marking corresponding to the new node is obtained by replacing $M(p)$ with ω whenever $M'(p) < M(p)$. Note that each ω generated in this way correctly identifies place unboundedness, as the sequence of transitions between the two nodes can be repeated indefinitely starting from M.

For PT-nets, the algorithm always terminates and the resulting *finite* coverability tree CT can be used to decide various relevant properties. In particular, one can show

Table 1. Algorithm generating a coverability tree of a PTI-net $N = (P, T, W, I, M_0)$

```
$CT = (V, A, \mu, v_0)$ where $V = \{v_0\}$, $A = \emptyset$ and $\mu(v_0) = M_0$
$unprocessed = \{v_0\}$
while $unprocessed \neq \emptyset$
    let $v \in unprocessed$
    if $\mu(v) \notin \mu(V \setminus unprocessed)$ then
    for every $\mu(v)[t\rangle M$
        $V = V \uplus \{w\}$ and $A = A \cup \{v \xrightarrow{t} w\}$
                and $unprocessed = unprocessed \cup \{w\}$
        if there is $u$ such that $u \rightsquigarrow_A^\sigma v$ and $\mu(u) < M$
            and $\mu(u)(p) < M(p)$ implies $I(t') = \emptyset$, for all transitions $t'$ in $\sigma t$    $(*)$
        then $\mu(w)(p) = ($if $\mu(u)(p) < M(p)$ then $\omega$ else $M(p))$
        else $\mu(w) = M$
    $unprocessed = unprocessed \setminus \{v\}$
```

that all reachable markings of N are covered by the extended markings associated with the nodes of CT. Moreover, for each extended marking M associated with a node of CT there are reachable markings $M_1, M_2, \ldots$ of N approximating M, i.e. for each n and every $p \in P$, $M_n(p) = M(p)$ if $M(p) \in \mathbb{N}$ and $M_n(p) \geq n$ otherwise. As an immediate consequence, CT can be used to decide unboundedness of places in reachable markings.

This standard coverability tree algorithm does not work for PTI-nets; it may in fact falsely identify places as unbounded. The problem is the non-monotonicity of PTI-nets: given two markings $M < M'$ and a firing sequence σ leading from M to M', it is not guaranteed that σ can also be fired from M'. As a consequence, the condition for generating ω-components may be too weak. It can be strengthened by ensuring that no inhibitor features were used along the path from u to v for those places where the number of tokens has grown. This has led to the modification proposed in [6], i.e. the addition of the line marked with $(*)$ in the algorithm in Table 1 which is the only difference with the standard coverability tree algorithm for PT-nets: the marking corresponding to the new node is obtained by replacing each $M(p)$ with ω provided that $M'(p) < M(p)$ **and** the transitions fired between the two nodes have no inhibitor places

In [6] it has been shown that the algorithm in Table 1 will always terminate for a PTI-net with one inhibitor place. Moreover, an example was given that this no longer holds if the net contains three such places. The termination problem in the case of exactly two inhibitor places was left open.

4 A Counterexample

In this section we will show an example of a PTI-net with two inhibitor places for which the algorithm in Table 1 does not terminate. An important insight in the design of this

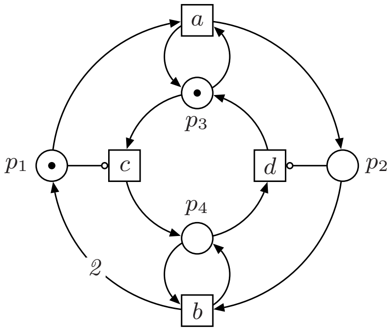

Fig. 1. PTI-net N_0 with two inhibitor places for which the algorithm in Table 1 does not terminate.

counterexample was that its two inhibitor places need to be simultaneously unbounded. This follows from the proof in [6] for PTI-nets with one inhibitor place. In short, if they would not grow simultaneously, one of the inhibitor places would generate an ω-component. The remainder of the construction would then regard the net as a PTI-net with a single inhibitor place, and terminate.

The counterexample is shown in Figure 1. We first observe that the two inhibitor places p_1 and p_2 of N_0 are simultaneously unbounded. It is easy to see that N_0 is deterministic and can fire exactly one infinite sequence of transitions $\sigma = \sigma_0\sigma_1\sigma_2\dots$, where $\sigma_i = a^{2^i}cb^{2^i}d$ for every $i \geq 0$. For any σ_i, the firing of transitions a, c and d does not change the total amount of tokens. However, each firing of transition b adds one token to the total count. Hence N_0 is unbounded and the inhibitor places, p_1 and p_2, are simultaneously unbounded (one only needs to consider the markings reached after firing transition sequences $\sigma_0\sigma_1\dots\sigma_i a^{2^i}$ for $i \geq 0$).

Theorem 1. *The algorithm in Table 1 does not terminate for PTI-net N_0 in Figure 1.*

Proof. Figure 2 shows part of the reachability graph N_0 (which has the shape of an infinite line). It starts with a marking M_1 having $x > 0$ tokens in place p_1, and one token in place p_3. There are three situations in which markings cover ancestor markings in this fragment.
Case 1: $M_1 = (x, 0, 1, 0)$ is covered by $M_7 = (2x, 0, 1, 0)$ and $M_8 = (2x - k, k, 1, 0)$ when $0 < k \leq x$. Between M_1 and M_7, M_8 transitions c and d fire. As $I(c), I(d) \neq \emptyset$, the line marked with an $*$ in the algorithm in Table 1 returns *false*, and so no ω components will be produced.
Case 2: $M_2 = (x - i, i, 1, 0)$ is covered by $M_8 = (2x - k, k, 1, 0)$ with $0 < k - i \leq x$. Between M_2 and M_8 transitions c and d fire, and so no ω components will be produced.
Case 3: $M_3 = (0, x, 1, 0)$ is covered by $M_8 = (2x - k, k, 1, 0)$ with $x < k < 2x$, and by $M_9 = (0, 2x, 1, 0)$. Between M_3 and M_8, M_9 transitions c and d fire, and so no ω components will be produced.

We now observe that the initial marking $M = (1, 0, 1, 0)$ is M_1 with $x = 1$, and that M_6 is M_1 with x replaced by $2x$. Consequently, the above argument can be applied to the entire infinite sequence of firings of N_0, with all possible coverings identified as in the above case analysis. As no covering leads to the generation of ω components, the algorithm will never terminate. □

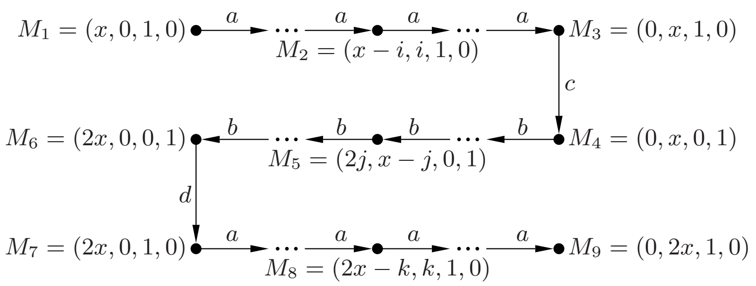

Fig. 2. Execution of the PTI-net in Figure 1 with $0 < i, j < x$ and $0 < k < 2x$.

5 Discussion

This article focused on the number of inhibitor places in a PTI-net. However, since an inhibitor place may inhibit multiple transitions, one can also focus on the number of inhibitor arcs in the PTI-net. Still, as our counterexample has only *two unweighted inhibitor arcs*, it also closes the gap between one inhibitor place and two inhibitor arcs. In this context it is worthwhile to notice that in [6] a construction is given which simulates the inhibitor arcs connected to a single inhibitor place by just one unweighted inhibitor arc.

We have seen that the Modified Coverability Tree Construction does not in general terminate for PTI-nets with two or more inhibitor places. This was to be expected, as the modelling power of such PTI-nets reaches Turing completeness. To have more decision power on such Petri net models, one needs to consider restricted subclasses. An example of such a subclass are the Primitive PTI-nets, introduced in [2].

References

1. Agerwala, T.: A Complete Model for Representing the Coordination of Asynchronous Processes. Tech. rep., Johns Hopkins University (1974)
2. Busi, N.: Analysis Issues in Petri Nets with Inhibitor Arcs. Theoretical Computer Science 275(1-2), 127 – 177 (2002)
3. Finkel, A.: A Generalization of the Procedure of Karp and Miller to Well Structured Transition systems. In: 14th International Colloquium on Automata, languages and programming. pp. 499–508. Springer-Verlag, London, UK (1987)
4. Hack, M.: Petri Net Languages. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA (1976)
5. Karp, R.M., Miller, R.E.: Parallel Program Schemata. Journal of Computer and System Sciences 3(2), 147 – 195 (1969)
6. Kleijn, J., Koutny, M.: Steps and Coverability in Inhibitor Nets. In: Lodaya, K., Mukund, M., Ramanujam, R. (eds.) Perspectives in Concurrency Theory. pp. 264–295. Universities Press, Hyderabad (2008)
7. Murata, T.: Petri nets: Properties, Analysis and Applications. In: Proceedings of the IEEE, Apr 1989, Volume: 77 Issue:4. pp. 541 – 580 (1989)
8. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall PTR, Upper Saddle River, NJ, USA (1981)