On Persistent Reachability in Petri Nets*

Kamila Barylska[1], Łukasz Mikulski[1], Edward Ochmanski[1,2]

[1] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Torun
[2] Institute of Computer Science, Polish Academy of Sciences, Warszawa, Poland
{khama,frodo,edoch}@mat.uni.torun.pl

Abstract. The notion of persistency, based on the rule "no action can disable another one" is one of the classical notions in concurrency theory. In this paper, we deal with arbitrary place/transition nets, but concentrate on their persistent computations. It leads to an interesting decision problem: Is a given marking reachable with a persistent run? In order to study the persistent-reachability problem we define a class of nets, called nonviolence nets. We show that inhibitor nets can be simulated by the nonviolence nets, and that reachability and coverability problems are undecidable in the class of the nonviolence nets. Then we prove more: nonviolence nets can be simulated by the inhibitor nets, thus they are computationally equivalent to Turing machines.

1 Introduction

An action of a concurrent system is said to be persistent if, whenever it becomes enabled, it remains enabled until executed. This classical notion, introduced by Karp/Miller [9], is one of the most frequently discussed issues in the Petri net theory (papers [1,2,3,6,8,11,12] a.m.o.). A net is said to be persistent if each of its actions is persistent. And most of the papers about persistency deal with this subclass of place/transition nets). In this paper, we deal with arbitrary place/transition nets, but concentrate on their persistent computations. It leads to an interesting persistent-reachability problems: Is a given marking reachable (coverable) with a persistent run?

It is well known that the classical versions of the problems (Is a given marking reachable (coverable) in a given place/transition net?) are decidable (coverability: Karp/Miller [9], Hack [8]; reachability: Mayr [13], Kosaraju [10]). In order to study the persistent-reachability problem we introduce a class of nets, called nonviolence nets (Definition 3.1). They differ from place/transition nets only by the execution rule. Namely, only persistent executions are permitted. We show that inhibitor nets can be simulated by nonviolence nets (Proposition 4.4). Using this fact we prove that the reachability and coverability problems are undecidable in the class of the nonviolence nets (Propositions 4.5 and 4.7, respectively). Then we prove more: nonviolence nets can be simulated by the inhibitor nets (Proposition 4.8), thus the both are computationally equivalent to Turing machines.

* The research supported by Ministry of Science and Higher Education of Poland – grant N N206 258035

Recent Advances in Petri Nets and Concurrency, S. Donatelli, J. Kleijn, R.J. Machado, J.M. Fernandes (eds.), CEUR Workshop Proceedings, ISSN 1613-0073, Jan/2012, pp. 373–384.

Many extensions of Petri nets are known to be Turing powerful: inhibitor nets, priority nets (Hack [7]), self-modifying nets (Valk [17]), for instance. There is also a Turing powerful model restricting the standard execution rules to maximal concurrent steps (Burkhard [4], see also Starke [16]). But all the models allow a fight for sharing resources (tokens), whereas our model works in a completely peaceful way.

In the concluding section we notice that the free-choice nonviolence nets are easy transformable to place/transition nets (not necessarily free-choice ones). Hence, the coverability and reachability problems are decidable in the class of the free-choice nonviolence nets.

2 Petri Nets – Basic Definitions

The set of non-negative integers is denoted by $\mathbb{N}$. Given a set X, the cardinality (number of elements) of X is denoted by $|X|$, the powerset (set of all subsets) by 2^X, the cardinality of the powerset is $2^{|X|}$. Multisets over X are members of $\mathbb{N}^X$, i.e. functions from X into $\mathbb{N}$. For convenience, if the set X is finite, multisets of $\mathbb{N}^X$ will be represented by vectors of $\mathbb{N}^{|X|}$.

2.1 Petri Nets and Their Computations

The definitions concerning Petri nets are mostly based on Desel/Reisig [5].

Net is a triple $N = (P,T,F)$, where:

- P and T are finite disjoint sets, of *places* and *transitions*, respectively;
- $F \subseteq P{\times}T \cup T{\times}P$ is a relation, called the *flow relation*.

For all $a \in T$ we denote: $^\bullet a = \{p \in P \mid (p,a) \in F\}$ – the set of *entries* to a
$a^\bullet = \{p \in P \mid (a,p) \in F\}$ – the set of *exits* from a

Petri nets admit a natural graphical representation. Nodes represent places and transitions, arcs represent the flow relation. Places are depicted by circles, and transitions by boxes. The set of all finite strings of transitions is denoted by T^*, the empty string is denoted by ε, the length of $w \in T^*$ is denoted by $|w|$, number of occurrences of a transition a in a string w is denoted by $|w|_a$.

Place/transition net (shortly, *p/t-net*) is a quadruple $S = (P,T,F,M_0)$, where:

- $N = (P,T,F)$ is a net, as defined above;
- $M_0 \in \mathbb{N}^P$ is a multiset of places, named the *initial marking*; it is marked by *tokens* inside the circles, capacity of places is unlimited.

Multisets of places are named *markings*. In the context of place/transition nets, they are mostly represented by nonnegative integer vectors of dimension $|P|$, assuming that P is strictly ordered. The natural generalizations, for vectors, of arithmetic operations + and –, as well as the partial order ≤, all defined componentwise, are well known and their formal definitions are omitted.

A transition $a \in T$ is *enabled* in a marking M whenever $^\bullet a \le M$ (all its entries are marked). If a is enabled in M, then it can be *executed*, but the execution is not forced. The execution of a transition a changes the current marking M to the new marking $M' = (M - {}^\bullet a) + a^\bullet$ (tokens are removed from entries, then put to exits). We shall denote: Ma for "a is enabled in M" and MaM' for "a is enabled in M and M' is the resulting marking". Then we say that MaM' is a *step*. This denotation we extend to strings of transitions: the empty string ε is enabled in any marking (always $M\varepsilon M$), a string $w=au$ ($a \in T$, $u \in T^*$) is enabled in a marking M whenever MaM' and u is enabled in M'. Predicates Mw and MwM' are defined like those for single transitions. If MwM' then we say that MwM' is a *computation* from M to M'. Note that any computation MwM' unambiguously defines all intermediate markings between M and M'.

If MwM', for some $w \in T^*$, then M' is said to be *reachable from M*. The set of all markings reachable from M is denoted by $[M\rangle$. Given a place/transition net $S=(P,T,F,M_0)$, the set $[M_0\rangle$ of all markings reachable from the initial marking M_0 is called the *reachability set* of S, and markings in $[M_0\rangle$ are said to be *reachable* in S.

We assume that the notions of *reachability* and *coverability graphs* are known to the reader. Their definitions can be found in any monograph or survey about Petri nets (see [5,16] or arbitrary else). Let us recall only that reachability graphs represent completely behaviours of nets, but are mostly infinite, while coverability graphs represent behaviours only partially, but are always finite. In Examples 2.2 and 2.3 we also use a notion of persistency graph – the reachability graph restricted to persistent steps.

2.2 Persistent Computations of Place/Transition Nets

The notion of persistency, proposed by Karp/Miller [9], belongs to the most important notions in concurrency theory. It is based on the behaviourally oriented rule "no action can disable another one", and generalizes the structurally defined notion of conflict-freeness.

Let $S=(P,T,F,M_0)$ be a place/transition net, and let M be a marking. The step MaM' is *persistent* iff $(\forall b \neq a)$ if Mb then $M'b$. The empty computation $M\varepsilon M$ is *persistent*; the computation $MaM'uM''$ is *persistent* iff the step MaM' is persistent and the computation $M'uM''$ is persistent. [In words: A computation is said to be *persistent* if any transition once enabled during this computation remains enabled until executed.] A p/t net is said to be *persistent* if it admits only persistent computations.

Example 2.1. *Non-persistent and persistent nets*

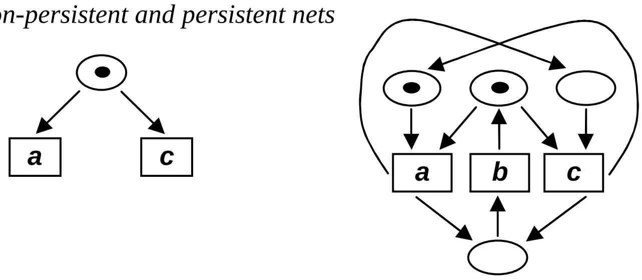

Fig. 1. A non-persistent (left) and persistent (right) place/transition nets

2.3 Persistent Reachability Problem

The problem of persistency ("Is a place/transition net persistent?"), raised by Landweber and Robertson in [11], has been proved to be decidable by Grabowski [6] and Mayr [12]. Most of p/t-nets, however, are not persistent, but some of their computations are persistent. In this paper, we are interested in markings that are reachable with persistent computations.

Let $S=(P,T,F,M_0)$ be a place/transition net, and let $M \in \mathbb{N}^P$ be a marking.
Reachability Problem: Is there a computation M_0wM?
In other words: Is the marking M reachable in the net S?

The Reachability Problem has been proved to be decidable by Mayr [13] and Kosaraju [10], after years of many author's efforts. A broad discussion, with a detailed proof, can be found in the book [15] of Reutenauer.

Let $S=(P,T,F,M_0)$ be a place/transition net, and let $M \in \mathbb{N}^P$ be a marking.
Persistent-Reachability Problem: Is there a persistent computation M_0wM?
In other words: Is the marking M reachable in the net S with a persistent run?

Obviously, if a p/t-net is persistent, then the persistent-reachability problem is equivalent to the classical one, thus decidable. We shall study the problem in general, for arbitrary p/t-nets. The following examples show difference between complete behaviours and persistent behaviours.

Example 2.2. *Comparison of the complete and persistent behaviours*

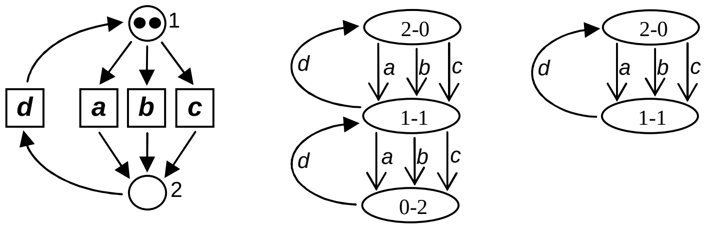

Fig. 2. A place/transition net and its reachability and persistency graphs

The above net is bounded (i.e. its reachability set is finite) and has infinite set of persistent computations. The example below shows an unbounded net (i.e. with infinite reachability set) with finite set (a singleton) of persistent computations.

Example 2.3. *Unbounded p/t-net with finite persistency graph*

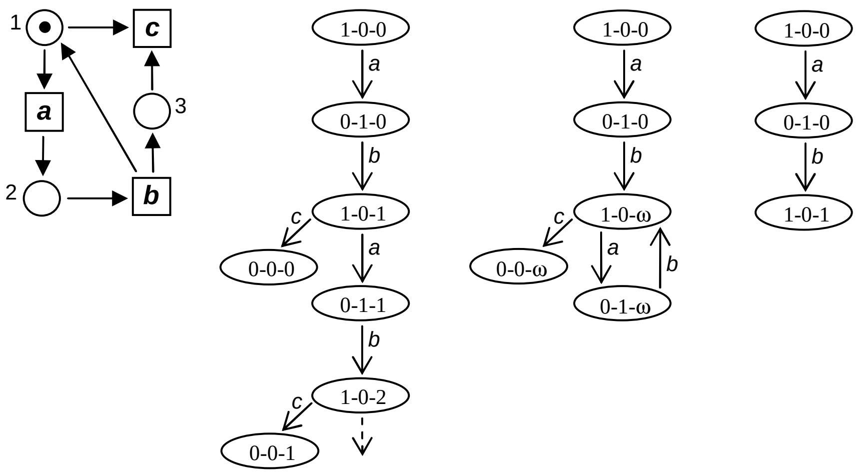

Fig. 3. A place/transition net and its reachability, coverability and persistency graphs

3 Nonviolence Petri Nets

In this section, we introduce the notion of nonviolence Petri nets. They differ from place/transition nets only by the execution rule. Namely, an enabled transition can be executed only if it is executable persistently (i.e. if its execution does not disable any other enabled transition). Therefore, we have to distinguish the notion "enabled" and "executable", that are synonymic in place/transition nets, but not in nonviolence nets.

Definition 3.1. *Nonviolence Petri Nets*
Nonviolence net is a quadruple $S = (P,T,F,M_0)$, exactly the same as in definition of place/transition nets. It differs from p/t-net by execution rules: A transition $a \in T$ is *enabled* in a marking M whenever ${}^\bullet a \leq M$ (all its entries are marked). A transition $a \in T$ is *executable* in M if it is enabled in M, and moreover the step MaM' is persistent. The execution of a leads to the resulting marking $M'=(M-{}^\bullet a)+a^\bullet$ (exactly same as in p/t-nets). We shall denote: $M\underline{a}$ for "a is executable in M" and $M\underline{a}M'$ for "a is executable in M and M' is the resulting marking". Then we say that $M\underline{a}M'$ is a *nonviolent step*. This denotation is naturally extended to strings $w \in T^*$. If $M\underline{w}M'$ then we say that $M\underline{w}M'$ is a *nonviolent computation*. Only nonviolent steps and computations are permitted in the nonviolence nets.

And now we can formulate the reachability and coverability problems for the nonviolence nets.

Let $S = (P,T,F,M_0)$ be a nonviolence net, and let $M \in \mathbb{N}^P$ be a marking.

NV-Reachability Problem:
Is there a nonviolence computation $M_0\underline{w}M$ in S?

NV-Coverability Problem:
Is there a marking $M' \geq M$ and a nonviolence computation $M_0\underline{w}M'$ in S?

3.2 From Place/Transition Nets to Nonviolence Nets

We shall show that every p/t-net can be simulated by a nonviolence net. It will be done by joining an external control to each transition of the net.

Let us consider an arbitrary p/t-net S. We transform it to the nonviolence net S' in the following way. To each transition a in the net S we join a switching transition a' and two new places p_a and q_a. We add the place q_a to the set of entries to a and to the set of exits from a'. We also add the place p_a to the set of exits from a and to the set of entries to a'. In initial marking we add one token to the place p_a. One can treat the constructed loop as a preparation of the transition a to execution.

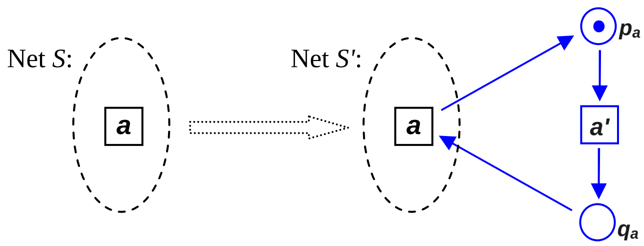

Fig. 4. Transforming a place/transition net into a nonviolence net

Let us also define, for every marking M in the net S, the marking $10M$ in the net S' as follows. For each place p in the net S we set $10M(p)=M(p)$, for each new place p_a we set $10M(p_a)$=1 and for each new place q_a we set $10M(q_a)$=0. With such definition, the initial marking in the net S' is $10M_0$, where M_0 is the initial marking in S. An obvious observation is that if transition a is executable in a marking M in the net S, then the sequence $a'a$ is executable in the marking $10M$ in the net S'.

Proposition 3.2. A marking M is reachable in a place/transition net S if and only if the marking $10M$ is reachable in the nonviolence net S'.
Proof. (⇒) Let M_0wM be a computation in S. Then replacing each a in w by $a'a$ we get a computation $10M_0\underline{w'}10M$ in the nonviolence net S'.
(⇐) Let $10M_0\underline{w'}10M$ in the nonviolence net S'. The only difference between behaviours of S and S' is that before every transition a a transition a' must be executed. Subsequent execution of two (or more) primed actions may sometimes disable the nonviolence execution of actions that were executable in S. However, it would not make any new action executable. Therefore, erasing all primed actions in $\underline{w'}$, we get a computation w such that M_0wM is a computation in the p/t-net S. □

4 Comparison of Nonviolence Nets and Inhibitor Nets

In this section, we recall the notion of inhibitor nets and some of their properties (undecidability of the the reachability and coverability problems). Then we show that their computational power is equal to that of the nonviolence nets.

Definition 4.1. *Inhibitor Petri Nets*
Inhibitor net is a quintuple $S = (P,T,F,I,M_0)$, where (P,T,F,M_0) is a place/transition net and $I \subseteq P\times T$ is the set of inhibitor arcs (depicted by edges ended with a small empty circle). Sets of entries and exits are denoted by $^\bullet a$ and $a^\bullet$, as in p/t-nets; the set of *inhibitor entries* to a is denoted by $^\circ a=\{p\in P \mid (p,a)\in I\}$.

A transition $a\in T$ is *enabled* in a marking M whenever $^\bullet a\leq M$ (all its entries are marked) and $(\forall p\in {^\circ a})\ M(p)=0$ – all *inhibitor entries* to a are empty. And “executable” means “enabled”, like in p/t-nets. The execution of a leads to the resulting marking $M'= (M-{^\bullet a})+a^\bullet$.

It is known that the inhibitor nets are computationally equivalent to Turing machines and the reachability problem in them is undecidable (Minsky [14], Hack [7]).

Fact 4.2. Reachability Problem is undecidable in the class of inhibitor nets.

Also the coverability problem is known to be undecidable in the class of inhibitor nets. We recall here the proving construction.

Let $S = (P,T,F,I,M_0)$ be an arbitrary inhibitor net with $P = \{p_1, \dots , p_k\}$, and let $M = [i_1, \dots , i_k]$ be a marking to be checked to be reachable. We extend it to an inhibitor net S', as follows: We add three new places p_0, p_{k+1}, p_{k+2} and two new transitions x, y, connected $p_0\rightarrow x\rightarrow p_{k+1}\rightarrow y\rightarrow p_{k+2}$ (see figure 5). Moreover, we join the place p_0 with every transition of the net S by a self-loop (it is depicted symbolically on Figure 5), we add the arcs from p_n to x, weighted by i_n (for n=1,…,k), [We use here the weighted arcs; see remark below.] and the inhibitory arcs from all original places of S to y. And the initial marking M'_0 in S' is the following: $M'_0(p_0)=1$, $M'_0(p_n)=M_0(p_n)$ for n=1,…,k and $M'_0(p_{k+1})=M'_0(p_{k+2})=0$.

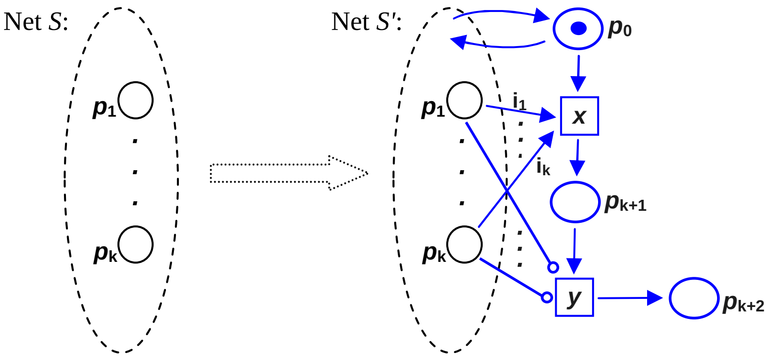

Fig. 5. Checking reachability with coverability in inhibitor nets

Remark. In this construction, we have used weighted (multiple) arcs. They are not mentioned, for simplicity, in our definitions; we assume that the notion is commonly known. Moreover, (place/transition or inhibitor) nets with multiple arcs can be transformed to the equivalent nets without them (Hack [8], Starke [16]).

Clearly, the marking $M = [i_1, \dots, i_k]$ is reachable in S if and only if the marking $M'(p_0)=M'(p_1)=\dots=M'(p_k)=M'(p_{k+1})=0$ and $M'(p_{k+2})=1$ is coverable in S'. Hence, because of Fact 4.2, we have got

Fact 4.3. Coverability Problem is undecidable in the class of inhibitor nets.

4.1 From Inhibitor Nets to Nonviolence Nets

Let us consider an arbitrary inhibitor net S. In the transformation to the nonviolence net S' we will use the same idea as in the previous transformation. Like in that one, to each transition a, we add a transition a' and places p_a and q_a. Moreover, to each place p, in the net S, belonging to $°a$ (i.e. being an inhibitor entry to a), we add a new transition a_p. Both places, p and q_a, are joined by self-loops with the new transition a_p. Finally, we remove all inhibitor arcs.

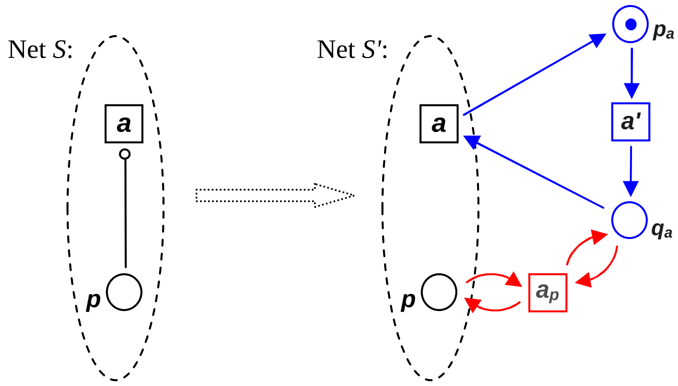

Fig. 6. Transforming an inhibitor net into a nonviolence net

Similarly to the construction of figure 4, for every marking M in the net S we define the marking $10M$ in the net S', in the same way. And the initial marking in the net S' is $10M_0$, where M_0 is the initial marking in the net S. In the same manner as in the previous case, if the transition a is executable in the marking M in the inhibitor net S, then the computation $a'a$ is executable in the marking $10M$ in the nonviolence net S'.

Proposition 4.4. The marking M is reachable in the inhibitor net S if and only if the marking $10M$ is reachable in the nonviolence net S'.
Proof. The proof is similar to that of Proposition 3.2. Remark that if an action a' is executed while a token resides in the place p (so a is not enabled in S), then a token will stuck in the place q_a and no marking of the form $10M$ will be reachable. □

Corollary 4.5. The NV-Reachability Problem is undecidable.
Proof. Directly from Proposition 4.4 and Fact 4.2. □

Proposition 4.6. The marking M is coverable in the inhibitor net S if and only if the marking $10M$ is coverable in the nonviolence net S'.
Proof. (⇒) If M is coverable in S then there is a marking $M' \geq M$ reachable in S. Hence, by Proposition 4.4, the marking $10M'$ is reachable in the nonviolence net S'. And clearly, $10M'$ covers $10M$.
(⇐) Notice that any marking (reachable in S') covering $10M$ is of the form $10M'$. And then $M' \geq M$ and M' is reachable in S (Proposition 4.4). So M is coverable in S. □

Corollary 4.7. The NV-Coverability Problem is undecidable.
Proof. Directly from Proposition 4.6 and Fact 4.3. □

4.2 From Nonviolence Nets to Inhibitor Nets

The inverse construction, transforming a nonviolence net into an inhibitor net, is more involved. Once more, we use the idea of predicting an executablement of transitions.

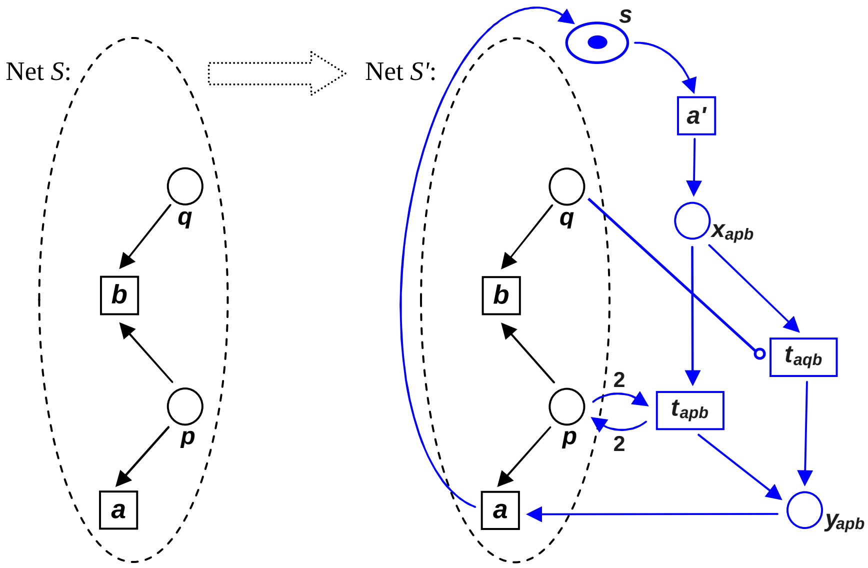

Fig. 7. Transforming a nonviolence net into an inhibitor net

Let *S* be a nonviolence net; we extend it to an inhibitor net *S'*, as follows. First, we add one global place *s*, called the switch, which is an exit from every transition of the net *S*. Then, for every transition *a* of the net *S*, we add a transition *a'*; the switch place *s* is an entry to each of these primed transitions. An execution of a transition *a'* means a belief in the executablement of transition *a* in the nonviolence net *S*. After executing the transition *a'*, the net *S'* checks, if the transition *a* was really executable in the nonviolence net *S*. It means that transition *a* is enabled and no other transition blocks its execution. In order to check it, we add, for every pair (*p*,*b*) such that $b \neq a$ and the place *p* is a common entry to the transitions *a* and *b*, the places x_{apb} and y_{apb} and the transition t_{apb}, as depicted on figure 7 above. The transition t_{apb} is able to move the token from x_{apb} to y_{apb} only if at least two tokens reside in the place *p*. [We use here the weighted arcs; see remark after figure 5.] By the nonviolence rules, a transition *b* blocks nothing if it is not enabled; it means that one of its entries is empty. In order to check it, we add the transitions t_{aqb}, for every entry *q* to transition *b*, different from *p*. Each of the transitions has one entry x_{apb}, one exit y_{apb} and one inhibitor arc from the place *q*, checking if *q* is empty. This construction allows to check, whether the transition *b* blocks the execution of *a* or not. If not, then a token moves from x_{apb} to y_{apb}, enabling *a* in *S'* if and only if it was executable in *S*.

For every marking *M* in the nonviolence net *S* we define a marking *10M* in the inhibitor net *S'* as follows. For each place *p* inside the net *S* we set *10M*(*p*)=*M*(*p*). For additional switch place s we set $10M(s)=1$ and for every place r added by the construction (i.e. for all places x_{apb} and y_{apb}) we set $10M(r)=0$. Directly from our construction, if a transition a is executable in a marking M in nonviolence net S then it is potentially executable in the marking $10M$ in inhibitor net S' (before executing a transition a we execute a transition a' and positively check all conditions to fill all places y_{apb}). The initial marking of S' is assumed to be $10M_0$.

Proposition 4.8. The marking M in the nonviolence net S is reachable if and only if the marking $10M$ is reachable in the inhibitor net S'.
Proof. (⇒) In the net S' we can reach marking $10M$, from the initial marking $10M_0$, by executing a transition a' before each transition a and checking the conditions.
(⇐) Executing any transition a from the original net S is possible only by predicting this execution by executing the transition a'. If we do a mistake, making wrong prediction, our net S' would reach a dead marking and stops. It means that if a marking $10M$ in the inhibitor net S' is reachable, then the only scenario of reaching that marking is correctly predicting and executing transitions from the net S. The correctness of our process of predicting means that we could just execute these transitions in the original, nonviolence net S, reaching marking M. Finally, marking M is reachable in the nonviolence net S, which ends the proof. □

Conclusions

We have proved (Propositions 4.4 + 4.8) that nonviolence nets are equivalent (in the marking reachability sense) to inhibitor nets. As the latter are Turing powerful, one can say that the former allow to do everything what possible without any fight. It is quite surprising, because persistent executions are only a part of arbitrary executions. But the price for the peace is undecidability. We have shown (Corollary 4.7) that even coverability, decidable in many extensions of place/transition nets, is undecidable in the class of the nonviolence nets.

Notice that free-choice (if $^\bullet a \cap {}^\bullet b \neq \emptyset$ then $^\bullet a = {}^\bullet b$) nonviolence nets can be simulated by place/transition nets (Figure 8), thus the classical decision problems (reachability, coverability) are decidable in the class of free-choice nonviolence nets.

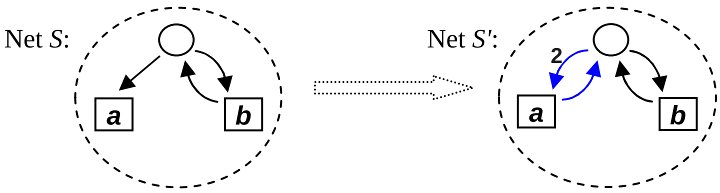

Fig. 8. Transforming a free-choice nonviolence net into a place/transition net

Let S be a free-choice nonviolence net. We replace every arc from a place, being a common entry of two (or more) transitions and is not a part of a self-loop, by two arcs: an arc from the place to the transition, weighted with 2, and an arc from the transition to the place, weighted with 1. And self-loops remain not changed. And the initial marking remains the same. Clearly, the place/transition net *S′* built this way works exactly as the free-choice nonviolence net *S*. A case of the free-choice nonviolence net is shown by Example 2.2. The above construction does not work for non-free-choice nonviolence nets, see Example 2.3, for instance.

It would be interesting to study some other subclasses of the class of nonviolence nets. Especially, to find a subclass of the class of nonviolence nets, computationally equivalent to the class of place/transition nets.

Acknowledgments

The paper was inspired by the Example 2.2. Great thanks are due to Ulla Goltz, who recalled to us that very old example.

References

1. E. Best: A Note on Persistent Petri Nets. Concurrency, Graphs and Models, LNCS 5065, pp. 427-438. Springer 2008.
2. E. Best, P. Darondeau: Decomposition Theorems for Bounded Persistent Petri Nets. Petri Nets 2008, LNCS 5062, pp. 33-51. Springer 2008.
3. E. Best, P. Darondeau: Separability in Persistent Petri Nets. Petri Nets 2010, LNCS 6128, pp. 246-266. Springer 2010.
4. H.-D. Burkhard: Ordered Firing in Petri Nets. Elektronische Informationsverarbeitung und Kybernetik 17(2/3), pp. 71-86, 1981.
5. J. Desel, W. Reisig: Place/Transition Petri Nets. Lectures on Petri Nets, LNCS 1491, pp. 122-173. Springer 1998.
6. J. Grabowski: The Decidability of Persistence for Vector Addition Systems. Information Processing Letters 11(1), pp. 20-23, 1980.
7. M. Hack: Petri Net Languages. Report MIT 124, 1975.
8. M. Hack: Decidability Questions for Petri Nets. Ph. D. Thesis, M.I.T. 1976.
9. R.M. Karp, R.E. Miller: Parallel program schemata. JCSS 3, pp. 147-195, 1969.
10. S.R. Kosaraju: Decidability of Reachability in Vector Addition Systems. Proc. of the 14th Annual ACM Symposium on Theory of Computing, pp. 267-281, 1982.
11. L.H. Landweber, E.L. Robertson: Properties of conflict-free and persistent Petri nets. Journal of ACM 25, pp. 352-364, 1978.
12. E. Mayr: Persistence of Vector Replacement Systems is Decidable. Acta Informatica 15, pp. 309-318, 1981.
13. E. Mayr: An algorithm for the general Petri net reachability problem. Proc. of the 13th Annual ACM Symposium on Theory of Computing, pp. 238-246, 1981.
14. M. Minsky: Computation: Finite and Infinite Machines. Prentice-Hall 1967.
15. C. Reutenauer: The Mathematics of Petri Nets. Prentice-Hall 1990.
16. P. H. Starke: Petri-Netze. VEB Berlin 1980.
17. R. Valk: Self-Modifying Nets, a Natural Extension of Petri Nets. ICALP 1978, LNCS 62, pp. 464-476. Springer 1978.