Complexity Results for 1-safe Nets

Allan Cheng[1], Javier Esparza[2], Jens Palsberg[1]

[1] Computer Science Dept., Aarhus University, Ny Munkegade, DK-8000 Århus C, Denmark, {acheng,palsberg}@daimi.aau.dk
[2] Laboratory for Foundations of Computer Science, University of Edinburgh, The King's Buildings, Edinburgh EH9 3JZ, UK, je@dcs.ed.ac.uk

Abstract. We study the complexity of several standard problems for 1-safe Petri nets and some of its subclasses. We prove that reachability, liveness, and deadlock are all PSPACE-complete for 1-safe nets. We also prove that deadlock is NP-complete for free-choice nets and for 1-safe free-choice nets. Finally, we prove that for arbitrary Petri nets, deadlock is equivalent to reachability and liveness.

1 Introduction

Petri nets are one of the oldest and most studied formalisms for the investigation of concurrency [30]. Shortly after the birth of complexity theory, Jones, Landweber, and Lien studied in their classical paper [22] the complexity of several fundamental problems for Place/Transition nets (called in [22] just Petri nets). Some years later, Howell, Rosier, and others studied the complexity of numerous problems for conflict-free nets, a subclass of Place/Transition nets [19, 20].

In the 1980's, process algebras were introduced as an alternative approach to the study of concurrency; they are more compositional and of higher level. The relationship between Petri Nets and process algebras has been thoroughly studied; in particular, many different Petri net semantics of process algebras have been described, see for instance [2, 7, 15, 29]. Also, a lot of effort has been devoted to giving nets an algebraic structure by embedding them in the framework of category theory, see among others [34, 26]. Although part of this work has been done for Place/Transition nets [15, 26], it has been observed that the nets in which a place can contain at most one token, called in the sequel *1-safe nets*, have many interesting properties. Places of 1-safe nets no longer model counters but logical conditions; a token in a place means that the corresponding condition holds. This makes 1-safe nets rather different from Place/Transition nets, even though both have similar representations; for instance, finite Place/Transition nets can have infinite state spaces, but finite 1-safe nets cannot.

The advantages of 1-safe nets are numerous, and they have become a significant model. Several semantics can be smoothly defined for 1-safe nets [3, 28], but are however difficult to extend to Place/Transition nets. Nielsen, Rozenberg and Thiagarajan [32, 28] have shown that a model of 1-safe nets, called Elementary Net Systems, has strong categorical connections with many other models of concurrency, such as event structures (another good reference is [35]). Finally,

Petri net classes	Reachability	Liveness	Deadlock
Arbitrary	decidable EXPSPACE-hard	decidable EXPSPACE-hard	decidable EXPSPACE-hard
1-safe	PSPACE-complete	PSPACE-complete	PSPACE-complete
Acyclic	NP-complete	linear time	linear time
1-safe acyclic	NP-complete	constant time	constant time
Conflict-free	NP-complete	polynomial time	polynomial time
1-safe conflict-free	polynomial time	polynomial time	polynomial time
Free-choice	decidable EXPSPACE-hard	NP-complete	NP-complete
1-safe free-choice	PSPACE-complete	polynomial time	NP-complete

Table 1. Summary of complexity results for Petri nets.

1-safe nets are closer to classical language theory, and can be interpreted as a synchronisation of finite automata.

These properties have motivated the design of verification methods particularly suited for 1-safe nets. Several different proposals have recently been presented in the literature [33, 14, 25, 10]. In order to evaluate them, and as a guide for future research, it is necessary to know the complexity of verification problems for 1-safe nets. This paper provides the first systematic study for 1-safe nets.

We study the maybe three most important verification problems for Petri nets: *reachability*, *liveness*, and existence of *deadlocks*. We determine their complexity for 1-safe nets, and for three important subclasses: *acyclic*, *conflict-free* and *free-choice* nets. In all cases, we compare the results with the complexity of the corresponding problems for Place/Transition nets.

This paper is a mixture of survey and new results. Our new results have enabled us to complete Table 1. Throughout, we attribute previously known results to their authors.

Two interesting subclasses of Petri nets are not covered by Table 1, namely S- and T-systems [4]. For those, reachability, liveness, and deadlock are known to be polynomial in the Place/Transition case [4, 6, 13], hence also in the 1-safe case. Related work concerning not the complexity of particular verification problems but the complexity of deciding different equivalence notions can be found in [21].

The paper is organised as follows. Section 2 contains basic definitions. In section 3 we show that the deadlock problem is recursively equivalent to the liveness and reachability problems. Section 4 shows that the three problems are PSPACE-complete in the 1-safe case. In section 5, the different classes of Petri nets mentioned above are considered. Several proofs are only sketched due to lack of space. The full versions can be found in [5].

We finish this introduction with a remark. In the paper, 1-safe nets are defined as a subclass of Place/Transition nets. Other versions of 1-safe nets can be found in the literature, namely the Condition/Event systems [30] and the

Elementary Net Systems [32]. This multiplicity of definitions is maybe annoying but harmless: the differences among them are small, and of rather technical nature (see [1] for a discussion). In particular, our results are independent of the definition used.

2 Definitions

We recall in this section some basic concepts about Place/Transition nets and 1-safe nets, and define the reachability, liveness and deadlock problems.

A *Place/Transition net*, or just a *net*, is a fourtuple $N = (P, T, F, M_0)$ such that

1. P and T are disjoint sets; their elements are called *places* and *transitions*, respectively.
2. $F \subseteq (P \times T) \cup (T \times P)$; F is called the *flow relation.*
3. $M_0: P \rightarrow \mathbb{N}$; M_0 is called the *initial marking* of N; in general, a mapping $M: P \rightarrow \mathbb{N}$ is called a *marking* of N

We assume that the reader is familiar with the following standard notation and notions: for $a \in P \cup T$ ${}^\bullet a$ $(a^\bullet)$ the *preset* (*postset*) of a, $t \in T$ being *enabled* (*occurring*) at a marking M, $\sigma = t_1 \cdots t_n$ a *firing sequence*, $[M\rangle$ the *reachable markings from* M.

Sometimes, we denote that a transition t has preset I and postset O in the following way: $t : I \rightarrow O$. For technical reasons we only consider nets in which every node has a nonempty preset or a nonempty postset. We will let $+$ denote union of sets and of multisets. Moreover, we will sometimes write a singleton set $\{p\}$ as simply p.

A marking M of a net N is *1-safe* if for every place p of the net $M(p) \leq 1$. We identify a 1-safe marking M with the set of places p such that $M(p) = 1$. A net N is *1-safe* if all its reachable markings are 1-safe.

A net N is *unary* if at every reachable marking at most one transition is enabled. N is *1-conservative* if for every transition t, $|{}^\bullet t| = |t^\bullet|$.

The *reachability problem* for a net N is the problem of deciding for a given marking M of N if it is reachable.

A net N is live if for every transition t of N and every reachable marking M, some marking of $[M\rangle$ enables t. The *liveness problem* for a net is the problem of deciding if it is live.

A marking of a net is a *deadlock* if it enables no transitions. The *deadlock problem* for a net is the problem of deciding if any of its reachable markings is a deadlock.

3 Place/Transition Nets

For Place/Transition nets, it is known that the liveness and reachability problems are recursively equivalent [16], and that they are both decidable and EXPSPACE-hard [23]. We complete the picture by showing that the deadlock problem is recursively equivalent to them, and thus decidable and EXPSPACE-hard.

Theorem 1. *Reachability is polynomial-time reducible to deadlock.*

Proof. Given a net $N = (P, T, F, M_0)$, and a marking M of N, we construct a net $N' = (P', T', F', M_0')$, as follows. Let V be the set of places marked in M. The places and transitions of N' are:

$$P' = P \cup \{p_t \mid t \in T\} \cup \{b_q, c_q \mid q \in V\}$$
$$T' = \{t_c \mid t \in T\} \cup \{t_p \mid p \in P\} \cup \{terminate\} \cup \{sub_q, loop_q \mid q \in V\}$$

The flow relation of N' is given by:

$$\begin{array}{lrcl}
\text{For each } t \in T: & t_c : {}^\bullet t + p_t & \rightarrow & t^\bullet + p_t \\
\text{For each } p \in P: & t_p : p & \rightarrow & p \\
 & terminate : \sum_{t \in T} p_t & \rightarrow & \sum_{q \in V} b_q \\
\text{For each } q \in V: & loop_q : c_q & \rightarrow & c_q \\
\text{For each } q \in V: & sub_q : c_q + q + b_q & \rightarrow & b_q
\end{array}$$

Finally, $M_0' = M_0 + \sum_{q \in V} \alpha_q c_q + \sum_{t \in T} p_t$ where $M = \sum_{q \in V} \alpha_q q$, $\alpha_q > 0$.

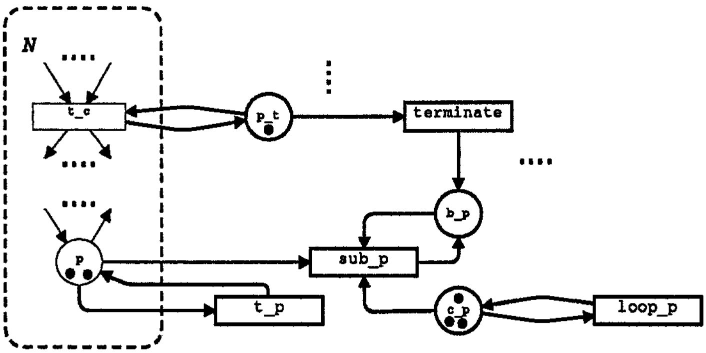

Fig. 1. Reducing reachability to deadlock.

The construction of N' is illustrated in figure 1.

Claim: M is reachable in N if and only if N' has a deadlock. To see this, first notice that *terminate* can occur at most once, that this disables all the t_c transitions, and that as long as it has not occurred, no marking can be dead: *terminate* can occur.

Suppose now that M is reachable in N. Having reached M in N' firing only t_c transitions, fire the *terminate* transition and use the sub_q transitions to remove, for each $q \in V$, α_q tokens from q. This yields a dead marking.

Suppose then that M is *not* reachable in N. Before *terminate* has fired, there is no deadlock. When *terminate* has fired, no transition in N can fire. There are two cases. Suppose first that M is the empty marking. Since M is not reachable in N, there are still tokens in N. Thus, at least one t_p transition will remain enabled. Suppose then that M is a non-empty marking. If there are no tokens in N, then at least one $loop_q$ transition will remain enabled. If there are still tokens in N, then at least one t_p transition will remain enabled.

Theorem 2. *Deadlock is polynomial-time reducible to liveness.*

Proof. Given a net $N = (P, T, F, M_0)$, we construct a net $N' = (P', T', F', M_0')$, as follows. The places and transitions of N' are:

$$P' = P \cup \{ok\}$$
$$T' = \{t_c, t' \mid t \in T\} \cup \{live\}$$

The flow relation of N' is given by:

$$\begin{array}{rl} \text{For each } t \in T\text{:} & t_c : {}^\bullet t \;\rightarrow\; t^\bullet \\ \text{For each } t \in T\text{:} & t' : {}^\bullet t \rightarrow ok \\ & live : ok \;\rightarrow\; P' \end{array}$$

Finally, $M_0' = M_0$.

Claim: N has no reachable dead marking if and only if N' is live.

Corollary 3. *The deadlock, liveness and reachability problems are recursively equivalent. Thus, the deadlock problem is decidable and EXPSPACE-hard.*

Proof. For the equivalence of the problems, combine theorems 1 and 2 with Hack's reduction from liveness to reachability [16]. For the complexity of the deadlock problem, use the equivalence with reachability and obtain the decidability from Mayr [24] and the EXPSPACE-hardness from Lipton [23].

4 1-Safe Nets

In this section we prove that the reachability, liveness and deadlock problems are PSPACE-complete for 1-safe nets. First we consider the liveness problem.

Theorem 4. *The liveness problem for 1-safe nets is PSPACE-complete.*

Proof. (Sketch) To prove that the liveness problem is in PSPACE, we can use essentially the technique of Jones, Landweber, and Lien [22, Theorem 3.9]. They proved that the liveness problem for 1-conservative (not necessarily 1-safe) nets is in PSPACE.

To prove completeness, we show that the problem (DETERMINISTIC) LINEAR BOUNDED AUTOMATON ACCEPTANCE (which is PSPACE-complete [12, p.265]) is polynomial-time reducible to the liveness problem. A linear bounded automaton is a Turing machine which only visits the cells of the tape containing the input. The input is bounded by a left and a right marker, say # and \$, and the head can visit no cell to the left of # and no cell to the right of \$ (see [18] for a formal definition). The problem is, given a deterministic linearly bounded automaton $\mathcal{M}$ and an input x for $\mathcal{M}$, to decide if $\mathcal{M}$ accepts x. We can further assume that $\mathcal{M}$ has a unique accepting configuration.

We construct a 1-safe net N which simulates $\mathcal{M}$. N contains a place for every pair of a tape cell and a tape symbol, and a place for every pair of a tape cell and a state. Given a configuration c of the automata $\mathcal{M}$, c can be encoded as a subset of places. The transitions change the distribution of tokens according to the transition function of $\mathcal{M}$.

Moreover, N contains two places B and C, which play the role of a switch, as follows. If there is a token on B, then the net simulates $\mathcal{M}$; if there is a token on C, then the net behaves nondeterministically in such a way that any marking corresponding to a configuration of the linear automaton can be reached. If the net reaches the marking corresponding to the accepting (initial) configuration of $\mathcal{M}$, then a transition can occur which moves the token from B to C (from C to B). Initially there is a token on B.

If $\mathcal{M}$ does not accept x, then the transition that moves the token from B to C can never occur, and therefore N is not live. If $\mathcal{M}$ accepts x, then this transition occurs. So C gets a token, and N starts behaving nondeterministically. Then, essentially, all behaviours are possible, and it follows that N is live.

We now consider the reachability problem. It is again possible to use a reduction from linear bounded automaton acceptance. However, we prefer to give another reduction from quantified boolean formulas. This reduction has some interest in itself, and moreover shows that the problem is still PSPACE-complete even if restricted to unary 1-safe nets.

Theorem 5. *The reachability problem for unary 1-safe nets is PSPACE-complete.*

Proof. (Sketch) The reachability problem is clearly in PSPACE: given a net N and a marking M of N, guess an occurrence sequence, and check in linear space that the occurrence sequence leads to M.

To prove PSPACE-hardness, we show that QUANTIFIED BOOLEAN FORMULAS (which is PSPACE-complete [12]) is polynomial-time reducible to the reachability problem.

If we are given a quantified boolean formula $\mathcal{F}$, then we construct a unary 1-safe net N and a marking M of N such that M is reachable iff $\mathcal{F}$ is true.

Before constructing the net and the marking, we rewrite $\mathcal{F}$, in polynomial time, into an equivalent closed formula G generated by the grammar:

$$P \ ::= \ x \mid \neg P \mid P \wedge P \mid \exists x.P$$

and such that all bound variables in G are distinct.

The net for G contains the places:

$$\{P\_in, P\_T, P\_F \mid P \text{ is an occurrence of a subformula of } G\} \cup$$
$$\{x\_is\_T, x\_is\_F \mid x \text{ is bound in } G\}$$

To avoid name clashes we could let the name of an occurrence of a subformula of G contain its position in the syntax tree for G.

The initial marking is $\{G\_in\}$.

The net for G contains the following transitions for each occurrence of a subformula of G:

Occurrence	Transitions
x	$\mathit{read\_x\_is\_T} : \mathit{x\_in} + \mathit{x\_is\_T} \rightarrow \mathit{x\_T} + \mathit{x\_is\_T}$
	$\mathit{read\_x\_is\_F} : \mathit{x\_in} + \mathit{x\_is\_F} \rightarrow \mathit{x\_F} + \mathit{x\_is\_F}$
$\neg P$	$\mathit{call\_P} : \mathit{not\_P\_in} \rightarrow \mathit{P\_in}$
	$\mathit{not\_P\_is\_F} : \mathit{P\_T} \rightarrow \mathit{not\_P\_F}$
	$\mathit{not\_P\_is\_T} : \mathit{P\_F} \rightarrow \mathit{not\_P\_T}$
$P \wedge Q$	$\mathit{call\_P} : \mathit{P\_and\_Q\_in} \rightarrow \mathit{P\_in}$
	$\mathit{P\_T\_and\_Q\_?} : \mathit{P\_T} \rightarrow \mathit{Q\_in}$
	$\mathit{P\_F\_and\_Q\_?} : \mathit{P\_F} \rightarrow \mathit{P\_and\_Q\_F}$
	$\mathit{P\_T\_and\_Q\_T} : \mathit{Q\_T} \rightarrow \mathit{P\_and\_Q\_T}$
	$\mathit{P\_T\_and\_Q\_F} : \mathit{Q\_F} \rightarrow \mathit{P\_and\_Q\_F}$
$\exists x.P$	$\mathit{call\_P\_with\_x\_T} : \mathit{Ex.P\_in} \rightarrow \mathit{P\_in} + \mathit{x\_is\_T}$
	$\mathit{call\_P\_with\_x\_F} : \mathit{x\_is\_T} + \mathit{P\_F} \rightarrow \mathit{x\_is\_F} + \mathit{P\_in}$
	$\mathit{x\_T\_P\_T} : \mathit{x\_is\_T} + \mathit{P\_T} \rightarrow \mathit{Ex.P\_T}$
	$\mathit{x\_F\_P\_T} : \mathit{x\_is\_F} + \mathit{P\_T} \rightarrow \mathit{Ex.P\_T}$
	$\mathit{Ex.P\_is\_F} : \mathit{x\_is\_F} + \mathit{P\_F} \rightarrow \mathit{Ex.P\_F}$

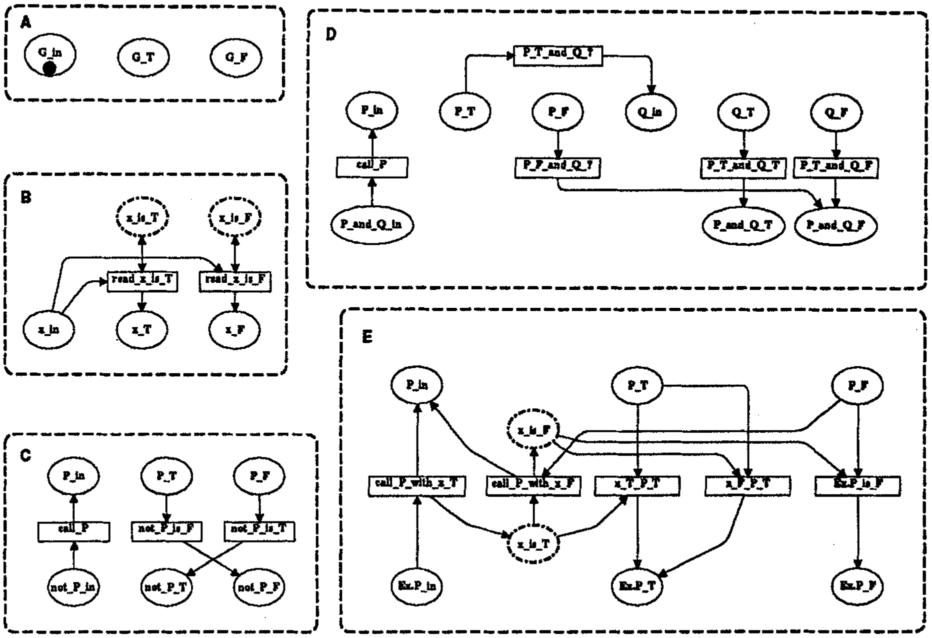

Fig. 2. Reduction from quantified boolean formulas.

The construction of the net for G is illustrated in figure 2. The idea is to try all possible assignments of bound variables. The construction is essentially compositional. The only complication is the interpretation of variables. Intuitively, when P_in ("the in-place for P") becomes marked, the checking of the truth of P begins. When either P_T ("true") or P_F ("false") becomes marked, this checking is completed.

Claim: $\{G\_T\}$ is reachable in the net for G if and only if G is true.

By a straightforward modification of the proof of Theorem 5, we get:

Theorem 6. *The deadlock problem for 1-safe nets is PSPACE-complete.*

The deadlock and and reachability problems turn out to be PSPACE-complete even for 1-conservative unary 1-safe nets, as shown in [5].

5 Subclasses

In this section we study the complexity of our three problems for three subclasses of nets which have been often studied in the literature. Most results are already known; we have collected them and filled some gaps. The nets of these subclasses satisfy some structural condition that rules out some basic kind of behaviours. In our first case, the *acyclic* nets, recursive or iterative behaviours are forbidden. The *conflict-free* nets do not allow nondeterministic behaviours (actually, this depends slightly on the notion of nondeterminism used). Finally, *free-choice* nets restrict the interplay between nondeterminism and synchronisations. In particular, in 1-safe free-choice nets the phenomenon of *confusion* [32] is ruled out.

5.1 Acyclic nets

A net $N = (P, T, F, M_0)$ is said to be acyclic if the transitive closure of F is irreflexive. The reachability problem remains untractable for acyclic 1-safe nets, although the problem is no longer PSPACE-complete (unless NP = PSPACE).

Theorem 7. *The reachability problem for acyclic 1-safe nets is NP-complete.*

Proof. The problem is in NP because in an occurrence sequence of a 1-safe acyclic net each transition occurs at most once. So we can guess an occurrence sequence in linear time and check in polynomial time if it leads to the given marking.

For the completeness part, see the paper by Stewart [31]. The result is proved by means of a reduction from the HAMILTONIAN CIRCUIT problem.

Since all 1-safe acyclic nets contain deadlocks, the liveness and deadlock problems are trivial. We then compare the results with those of the general case.

Theorem 8. *The reachability problem for acyclic Place/Transition nets is NP-complete.*

Proof. The problem can be polynomial-time reduced to INTEGER LINEAR PROGRAMMING, because in an acyclic net N with initial marking M_0, a marking M is reachable iff the system of equations corresponding to the state equation $M = M_0 + C \cdot X$, where C is the incidence matrix of N, has an integer vector solution X (for the definitions of incidence matrix and state equation, see, for instance, [27]). Since INTEGER LINEAR PROGRAMMING is in NP [18], so is our problem.

The completeness follows trivially from the completeness of the problem for the 1-safe case.

It is easy to see that an acyclic net has no deadlocks if and only if some of its transitions has empty preset; therefore the deadlock problem can easily be solved in linear time. Similarly, an acyclic net is live if and only if every place has some input transition; so the liveness problem is also linear. So, as we can see, there are no essential differences between the general and the 1-safe case.

5.2 Conflict-free nets

Conflict-free nets are a subclass in which conflicts are structurally ruled out (actually, this depends slightly on the notion of conflict used). Their complexity has been deeply studied in several papers; in particular, the complexity of our three problems.

A net $N = (P, T, F, M_0)$ is *conflict-free* if for every place p, if $|p^\bullet| > 1$, then $p^\bullet \subseteq {}^\bullet p$.

It is shown by Howell and Rosier in [19, 20] that the reachability, liveness, and deadlock problems for 1-safe conflict-free nets are solvable in polynomial time. They also show that, for conflict-free Place/Transition nets, the deadlock and liveness problems are still polynomial, whereas the reachability problem becomes NP-complete [19, 20].

5.3 Free-Choice nets

Free-choice nets are a well studied class, commonly acknowledged to be about the largest class having a nice theory.

A net $N = (P, T, F, M_0)$ is *free-choice* if for any pair $(p, t) \in F \cap (P \times T)$ it is the case that $p^\bullet = \{t\}$ or ${}^\bullet t = \{p\}$.

In a free-choice net, if some transitions share an input place p, then p is their unique input place. It follows that if any of them is enabled, then all of them are enabled. Therefore, it is always possible to freely choose which of them occurs.

The reachability problem is still PSPACE-complete for 1-safe free-choice nets. The reason is that for a 1-safe net N and a marking M, we can construct a 1-safe free-choice net N' containing all the places of N (and possibly more), such that M is reachable in N if and only if it is reachable in N'. N' is the so called 'released form' of N. Intuitively, every arc (p, t) such that $|p^\bullet| > 1$ and $|{}^\bullet t| > 1$ is removed and replaced by new arcs $(p, t'), (t', p'), (p', t)$, where p' and t' are a new place and a new transition. A formal definition is given in [22, 17].

Perhaps surprisingly, the liveness problem for 1-safe free-choice nets is polynomial, as shown by Esparza and Silva [11] and by Desel [8].

We now show that the deadlock problem for 1-safe free-choice nets is NP-complete. Membership in NP is non-trivial, and requires to introduce some concepts and results of net theory.

Theorem 9. *The deadlock problem for 1-safe free-choice nets is NP-complete.*

Proof. To prove membership in NP, we use the notions of siphon and trap. A subset Q of places of a net N is a *siphon* if ${}^\bullet Q \subseteq Q^\bullet$, and a *trap* if $Q^\bullet \subseteq {}^\bullet Q$. It is shown in [5] that a free choice net N has a deadlock iff there exists a siphon Q of N such that:

- for every transition t of N, Q contains some place of ${}^\bullet t$, and
- Q contains no trap marked at the initial marking (a trap is marked if at least one of its places is marked)

To solve the problem in nondeterministic polynomial time, we guess for each transition t of the net a place of ${}^\bullet t$, and then check in polynomial time if the guessed set of places is a siphon and if it contains no trap marked at the initial marking.

We prove completeness by reducing the satisfiability problem of propositional formulas in conjunctive normal form (CON-SAT) to the deadlock problem. Given an instance ϕ of CON-SAT, we construct a free-choice net N in polynomial time and show that that it has a deadlock iff ϕ is satisfiable. The construction is very similar to the one used in [22] to prove the NP-completeness of liveness in general free-choice nets.

There are differences between the 1-safe and the Place/Transition free-choice nets. Using the releasing technique it is easy to show that the reachability problem for free-choice nets is as hard as the reachability problem for arbitrary Place/Transition nets, and therefore EXPSPACE-hard. The liveness problem was shown to be NP-complete in [22]. Finally, our proof of membership in NP for the deadlock problem does not rely on 1-safeness; therefore, the deadlock problem is also NP-complete for Place/Transition free-choice nets.

6 Conclusions

We have analysed the complexity of several problems for 1-safe nets. Table 1 summarises results on the complexity of reachability, liveness, and existence of deadlocks. We can obtain some conclusions:

- All problems remain intractable, although, as could be expected, their complexity decreases in comparison with Place/Transition nets. The usual pattern is that problems are EXPSPACE-hard for Place/Transition nets and PSPACE-complete in the 1-safe case.
- Most problems remain intractable even for unary 1-safe nets, which are sequential and deterministic. So it is not possible to relate intractability to nondeterminism or concurrency.
- Some problems become tractable when restricted to subclasses of 1-safe nets defined using *structural* constraints, i.e., constraints on the flow relation.

The most interesting direction for further research is probably the study of the complexity of a problem when a certain desirable property is known to hold, for instance liveness. The result of [9] can be seen as a first step in this direction: it is shown that for live and 1-safe free-choice nets the reachability problem is in NP, by proving that every reachable marking can be reached by an occurrence sequence of polynomial length. So far nothing is known about the complexity of deciding if a marking is reachable when the Petri net is known to be live.

Acknowledgement. The authors thank Claus Torp Jensen for comments that lead to a simplification of the proof Theorem 5. The authors also thank S. Purushothaman, P.S. Thiagarajan, and the anonymous FST&TCS referees for helpful comments on a draft of the paper.

References

1. Luca Bernardinello and Fiorella De Cindio. A survey of basic net models and modular net classes. In *Advances in Petri Nets 1992*, pages 304–351. Springer-Verlag (*LNCS* 609), 1992.
2. Eike Best, Raymond Devillers, and Jon G. Hall. The Box calculus: a new causal algebra with multi-label communication. In *Advances in Petri Nets 1992*, pages 21–69. Springer-Verlag (*LNCS* 609), 1992.
3. Eike Best and César Fernández. *Nonsequential Processes – a Petri Net View.* EATCS Monographs on Theoretical Computer Science Vol.13, 1988.
4. Eike Best and P.S. Thiagarajan. Some classes of live and save Petri nets. In K. Voss, H.J. Genrich, and G. Rozenberg, editors, *Advances in Petri Nets*, pages 71–94. Springer-Verlag, 1987.
5. Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. Technical Report DAIMI PB-455, Computer Science Department, Aarhus University, September 1993.
6. Fred Commoner, Anatole W. Holt, S. Even, and Amir Pnueli. Marked directed graphs. *Journal of Computer and System Sciences*, 5:511–523, 1971.
7. Pierpaolo Degano, Rocco De Nicola, and Ugo Montanari. A distributed operational semantics for CCS based on C/E systems. *Acta Informatica*, 26:59–91, 1988.
8. Jörg Desel. A proof of the rank theorem. In *Advances in Petri Nets 1992*, pages 304–351. Springer-Verlag (*LNCS* 609), 1992.
9. Jörg Desel and Javier Esparza. Shortest paths in reachability graphs. In *Proc. Application and Theory of Petri Nets*, pages 224–241. Springer-Verlag (*LNCS* 691), 1993.
10. Javier Esparza. Model checking using net unfoldings. In *Proc. TAPSOFT'93*, pages 613–628. Springer-Verlag (*LNCS* 668), 1993.
11. Javier Esparza and Manuel Silva. A polynomial-time algorithm to decide liveness of bounded free choice nets. *Theoretical Computer Science*, 102:185–205, 1992.
12. Michael R. Garey and David S. Johnson. *Computers and Intractability.* Freeman, 1979.
13. Hartmann J. Genrich and Kurt Lautenbach. Synchronisationsgraphen. *Acta Informatica*, 2:143–161, 1973.
14. Patrice Godefroid. Using partial orders to improve automatic verification methods. In *Proc. CAV'90, 2nd Workshop on Computer-Aided Verification*, pages 176–185. Springer-Verlag (*LNCS* 531), 1990.
15. Ursula Goltz. On representing CCS programs by finite Petri nets. In *Proc. MFCS'88, Mathematical Foundations of Computer Science*, pages 339–350. Springer-Verlag (*LNCS* 324), 1988.
16. Michel Hack. The recursive equivalence of the reachabilty problem and the liveness problem for Petri nets and vector addition systems. In *Proc. 15th Annual Symposium on Switching and Automata Theory*, pages 156–164, 1974.

17. Michel Hack. The equality problem for vector addition systems is undecidable. *Theoretical Computer Science*, 2:77–95, 1976.
18. John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publishing Company, 1979.
19. Rodney R. Howell and Louis E. Rosier. Completeness results for conflict-free vector replacement systems. *Journal of Computer and System Sciences*, 37:349–366, 1988.
20. Rodney R. Howell and Louis E. Rosier. Problems concerning fairness and temporal logic for conflict-free Petri nets. *Theoretical Computer Science*, 64(3):305–329, 1989.
21. Lalita Jategaonkar and Albert Meyer. Deciding true concurrency equivalences on finite safe nets. In *Proc. ICALP'93*, pages 519–531, 1993.
22. Neil D. Jones, Lawrence H. Landweber, and Y. Edmund Lien. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 4:277–299, 1977.
23. Richard J. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.
24. Ernst W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal on Computing*, 13:441–460, 1984.
25. Kenneth L. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In *Proc. CAV'92, Fourth Workshop on Computer-Aided Verification*, pages 164–174, 1992.
26. José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and Computation*, 88:105–155, 1990.
27. Tadao Murata. Petri nets: Properties, analysis and applications. In *Proc. of the IEEE*, volume 77(4), pages 541–580, 1989.
28. Mogens Nielsen, Grzegorz Rozenberg, and P.S. Thiagarajan. Behavioural notions for elementary net systems. *Distributed Computing*, 4(1):45–57, 1990.
29. Ernst R. Olderog. *Nets, Terms and Formulas*. Cambridge University Press, 1991. *Number 23 Tracts in Theoretical Computer Science.*
30. Wolfgang Reisig. *Petri Nets – An Introduction*. EATCS Monographs in Computer Science Vol.4, 1985.
31. Iain A. Stewart. On the reachability problem for some classes of Petri nets. Research Report, University of Newcastle upon Tyne, 1992.
32. P.S. Thiagarajan. Elementary net systems. In *Advances in Petri Nets 1986, part I*, pages 26–59. Springer-Verlag (*LNCS* 254), 1987.
33. Antti Valmari. Stubborn sets for reduced state space generation. In Grzegorz Rozenberg, editor, *Advances in Petri Nets 1990*, pages 491–515. Springer-Verlag (*LNCS* 483), 1990.
34. Glynn Winskel. Petri nets, algebras, morphisms and compositionality. *Information and Computation*, 72(3):197–238, 1987.
35. Glynn Winskel and Mogens Nielsen. Models for concurrency. Technical Report DAIMI PB–429, Computer Science Department, Aarhus University, November 1992. To appear as a chapter in the Handbook of Logic and the Foundations of Computer Science, Oxford University Press.