Complexity Results for 1-safe Nets

Allan Cheng
Computer Science Department, Aarhus University
Ny Munkegade, DK-8000 Aarhus C, Denmark
acheng@daimi.aau.dk

Javier Esparza
Laboratory for Foundations of Computer Science
University of Edinburgh, The King's Buildings
Edinburgh EH9 3JZ
je@dcs.ed.ac.uk

Jens Palsberg
Computer Science Department, Aarhus University
Ny Munkegade, DK-8000 Aarhus C, Denmark
palsberg@daimi.aau.dk

September 1993

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.

This paper is to be presented at *FST&TCS 13, Foundations of Software Technology & Theoretical Computer Science*, to be held 15-17 December 1993, in Bombay, India. A version of the paper with most proofs omitted is to appear in the proceedings.

1 Introduction

Petri nets are one of the oldest and most studied formalisms for the investigation of concurrency [33]. Shortly after the birth of complexity theory, Jones, Landweber, and Lien studied in their classical paper [24] the complexity of several fundamental problems for Place/Transition nets (called in [24] 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 [21][22].

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 [3][7][16][32]. 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 [38][29]. Although part of this work has been done for Place/Transition nets [16][29], 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 iniinite 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 [4][31], but are however difficult to extend to Place/Transition nets. Nielsen, Rozenberg and Thiagarajan [36][31] 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

[39]). Finally, 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 [37][15][28][11]. 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.

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

Figure 1: Summary of complexity results for Petri 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. In a brief final section we study some other problems of interest.

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 [36]. For those, reachability, liveness, and deadlock are known to be polynomial in the Place/Transition case [36][6][14], 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 [23].

The paper is organised as follows. Section a 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. Finally, in section 6 other problems are studied.

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 [33] and the Elementary Net Systems [36]. 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 *transition*, respectively.
2. $F \subseteq (P \times T) \cup (T \times P)$; F is called the *flow relation*.
3. $M_0 : P \rightarrow I\!N; M_0$ is called the *initial marking* of N; in general, a mapping $M : P \rightarrow I\!N$ is called a *marking* of N

Given $a \in P \cup T$, the *preset* of a, denoted by ${}^\bullet a$, is defined as $\{a' \mid a'Fa\}$; the *postset* of a, denoted by $a^\bullet$, is defined as $\{a' \mid aFa'\}$.

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 multisets.

Let $N = (P, T, F, M_0)$ be a net. A transition $t \in T$ is *enabled* at a marking M of N if $M(p) > 0$ for every place p in the preset of t. Given a transition t, we define a relation $\xrightarrow{t}$ between markings as follows: $M \xrightarrow{t} M'$ if t is enabled at M and $M'(s) = M(s) + F(t, s) - F(s, t)$, where $F(x, y)$ is 1 if $(x, y) \in F$ and 0 otherwise. The transition t is said to *occur* (or *fire*) at M. If $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} M_n$ for some markings $M_0, M_1 \ldots M_n$, then the sequence $\sigma = t_1 \ldots t_n$ is called an *occurence sequence*. M_n is the marking *reached* by σ, and this is denoted $M_0 \xrightarrow{\sigma} M_n$. A marking M is *reachable* if it is the marking reached by some occurrence sequence. Given a marking M of N, the set of reachable markings of the net (P, T, F, M) (i.e., the net obtained replacing the initial marking M_0 by M) is denoted by $[M\rangle$.

Notice that the empty sequence is an occurrence sequence and that it reaches the initial marking M_0.

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-conseruative* 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 [18], and that they are both decidable and EXPSPACE-hard [26]. 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:

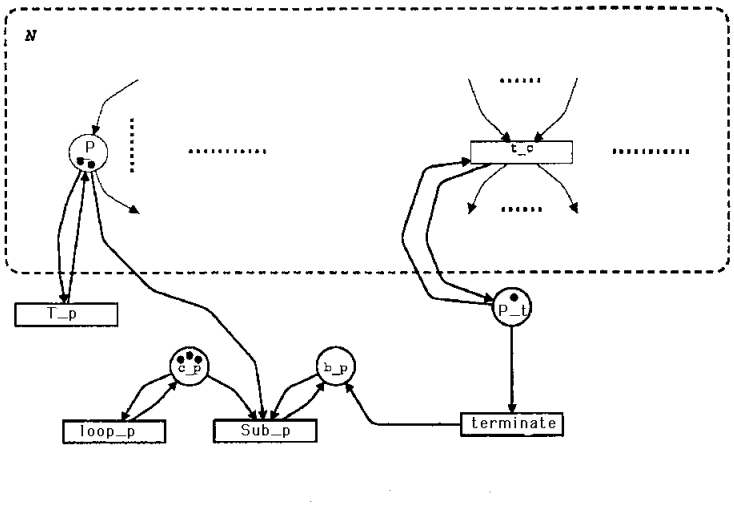

Figure 2: Reducing reachability to deadlock.

The flow relation of N' is given by:

For each $t \in T$: $t_c : {}^\bullet t + p_t \to t^\bullet + p_t$
For each $p \in P$: t_p $: p \to p$

$$
\begin{array}{lll}
 & terminate & : \sum_{t \in T} p_t \rightarrow \sum_{q \in V} b_q \\
\text{For each } q \in V\text{:} & loop_q & : c_q \rightarrow c_q \\
\text{For each } q \in V\text{:} & sub_q & : c_q + q + b_q \rightarrow b_q
\end{array}
$$

Finally,

$$M_0' = M_0 + \sum_{q \in V} \alpha_q c_q + \sum_{t \in T} p_t$$

where

$$M = \sum_{q \in V} \alpha_q q, \quad \alpha_q > 0$$

The construction of N' is illustrated in Figure 2.

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:
For each $t \in T$: t_c : ${}^\bullet t \to t^\bullet$
For each $t \in T$: t' : ${}^\bullet t \to ok$
live : $ok \to P'$
Finally, $M_0' = M_0$.

Claim: N has no reachable dead marking if and only if N' is live. To see this, suppose first that N can reach a dead marking M_d. Clearly, also N' can reach M_d without firing any t' transitions, and since the t' transitions in N' have the same presets as the transitions in N, M_d is dead in M'. Thus, N' is not live.

Suppose then that N has no reachable dead marking. Then the initial marking is not dead, so fire one of the t' transitions. This places a token on the *ok* place, and there the token remains. Thus from now on, the *live* transition is enabled, and because the *live* transition places tokens on *all* places in N', N' is live.
□

Corollary 3 *The deadlock, liveness and reachability problems are recurively 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 [18]. For the complexity of the deadlock problem, use the equivalence with reachability and obtain the decidability from Mayr [27] and the EXPSPACE-hardness from Lipton [26].
□

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. To prove that the liveness problem is in PSPACE, we can use essentially the technique of Jones, Landweber, and Lien [24, 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 [13, pp.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 [20] for a formal definition).
The problem is deiined as follows:

> Given: a deterministic linearly bounded automaton $\mathcal{M}_0$ and an input x for $\mathcal{M}_0$,
> To decide: if $\mathcal{M}_0$ accepts x.

First, we construct in polynomial time a deterministic linearly bounded automaton M, satisfying the following two properties:

> (1) $\mathcal{M}$ accepts x iff $\mathcal{M}_0$ accepts x, and
> (2) $\mathcal{M}$ has a unique accepting configuration.

$\mathcal{M}$ simulates $\mathcal{M}_0$, but, before accepting, $\mathcal{M}$ erases the tape, moves the head to the leftmost cell, and then enters its unique final state (a new state not present in $\mathcal{M}_0$). In this way, $\mathcal{M}$ satisfies (2).

Let $\mathcal{M} = (K, \sum, \Gamma, \delta, q_1, q_2, \#, \$)$ where K is the set of states, $\sum$ the alphabet,$\Gamma \supseteq \{\#, \$\}$ is the set of tape symbols, δ is the transition relation, q_1 the initial state, q_2 the final state, and # and $ are the boundary symbols. Moreover, let $K = \{q_1, \ldots, q_m\}, \Gamma = \{a_1, \ldots, a_p\}$, n = the size of $\#x\$$, and $\beta = K \times \Gamma \times \{C, R, L\} \times K \times \Gamma$ (i.e., the transition relation is a subset of β).

We construct a 1-safe net $N = (P, T, F, M_0)$ as follows:

- P = $\{A_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq p\}$
  $\cup \{Q_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$
  $\cup \{B, C\}$

  P contains a place $A_{i,j}$ for every tape cell i and every tape symbol a_j; a token in $A_{i,j}$ means that the symbol on tape cell i is a_j. It also contains a place $Q_{i,j}$ for every tape cell i and every state q_j; a token in $Q_{i,j}$ means that the automaton scans the cell i in state q_j. Given a configuration c of the automata $\mathcal{M}$, c can be encoded as a subset of P in the following way:

  - if the automaton is in state q_j scanning the i-th tape cell, then $Q_{i,j}$ belongs to the set,
  - if the tape cell i contains the symbol a_j, then $A_{i,j}$ belongs to the set, and
  - no other place belongs to the set.

  Denote the set of places associated to the configuration c by $M(c)$. Notice that $M(c)$ can also be interpreted as a 1-safe marking of N.

  B and C play the role of a switch, as follows. If there is a token on B, then the net simulates 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.

- T contains the following transitions for every element of β:

  - If $(q_s, a_t, R, q_r, a_l) \in \delta$ (move right), then T includes for every cell $1 \leq i < n$ a transition

    $$Q_{i,s} + A_{i,t} \to Q_{i+1,r} + A_{i,l}$$

    (where we use + instead of set union to use the notation of [24]; notice that no transition is needed for the n-th cell). Similarly for left moves and no motion. The transitions corresponding to an element of $\beta \setminus \delta$ have C in their preset, and can therefore only occur if C is marked.
  - If $(q_s, a_t, R, q_r, a_l) \in \beta \setminus \delta$, then T includes for every cell $1 \leq i < n$ a transition

    $$C + Q_{i,s} + A_{i,t} \to Q_{i+1,r} + A_{i,l} + C$$

Similarly for left moves and no motion.

- T contains the following two transitions $t_{B\to C}, t_{C\to B}$ where c_i is the initial configuration of $\mathcal{M}$, and C_f its unique accepting configuration.

  $$t_{B\to C} : B + M(c_f) \to C + M(c_f)$$

  If the net reaches the marking corresponding to the accepting configuration c_f then the transition $t_{B\to C}$ can occur and the net starts behaving nondeterministically in such a way that for any configuration c, the marking $C + M(c)$ is reachable.

  $$t_{C\to B} : C + M(c_i) \to B + M(c_i)$$

  The net can return to simulating $\mathcal{M}$ if, while behaving nondeterministically, it reaches the marlcing corresponding to the initial configuration.

- The initial marking M_0 is the one corresponding to the initial configuration, plus one token on the place B i.e., $M_0 = B + M(c_i)$

If $\mathcal{M}$ does not accept x, then N never reaches the marking $B + M(c_f)$, corresponding to the accepting configuration c_f. This implies that the transition $t_{B\to C}$ can never occur, and therefore N is not live.

If $\mathcal{M}$ accepts x, then the net reaches the accepting configuration c_f. So the transition $t_{B\to C}$ can occur, and N starts behaving nondeterministically. Now, for every possible configuration c, the net can reach $C + M(c)$. Hence every transition, but $t_{B\to C}$, can become enabled at some reachable marking containing C. In particular, the marking $M(c_i) + C$ can be reached too; this marking enables $t_{C\to B}$. Therefore, the net can return to simulating $\mathcal{M}$, and everything starts anew, in particular $t_{B\to C}$ can occur again.
□

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. 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 [13]) is polynomial-time reducible to the reachability problem.

The problem is defined as follows:

> Given: A well-formed quantified Boolean formula
>
> $$\mathcal{F} = (Q_1, x_1)(Q_2, x_2) \ldots (Q_n, x_n)E$$
>
> where E is a Boolean expression involving the variables $x_1, x_2, \ldots, x_n$ and each Q_i is either "$\exists$" or "$\forall$".

To decide: is $\mathcal{F}$ true?

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 if and only if $\mathcal{F}$ 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. Notice that G needs not be a quantified boolean formula: the quantifiers in G need not occur at the outermost level.

The construction of the net for G is illustrated in Figure 2. Intuitively, the idea is to try all possible assignments of bound variables. The construction is essentially compositional. The only complication is the interpretation of variables.

The net for G contains the places:

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

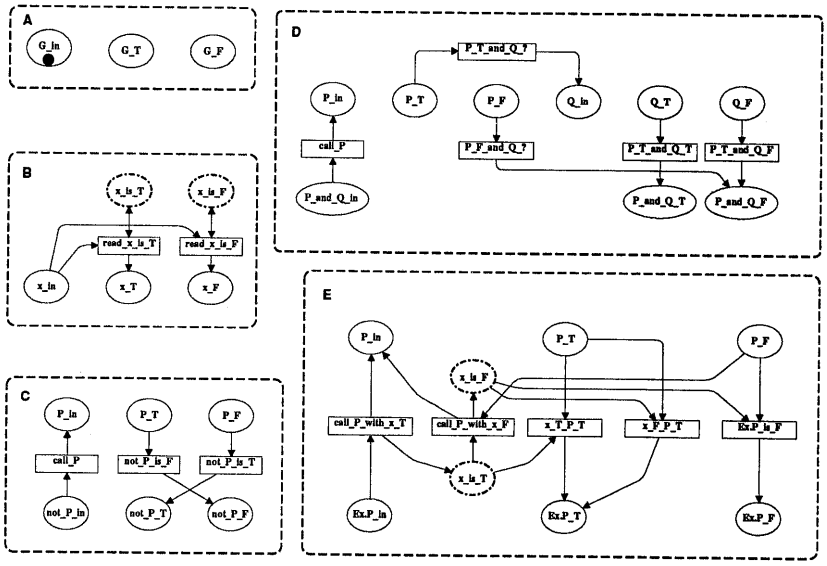

Figure 3: Reduction from quantified boolean formulas

For readability, when in the following we name places and transitions, we write not_P for $\neg P$, we write P_and_Q for $P \wedge Q$, and we write $Ex.P$ for $\exists x.P$.

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

The net for G contains the following transitions for each occurrence of a subformula of 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. We omit these details, for readability.

Intuitively, when P_in ("the in-place for P") becomes marked, then the checking of the truth of P begins. When either P_T ("true") or P_F ("false") becomes marked, this checking is completed. Let us consider in turn the construction for each of the productions of the above grammar.

First, consider a variable x, see Figure 3, box B. The places x_is_T ("x is true") and x_is_F ("x is false") are not part of the net for x but are included to indicate that they will be added when treating the quantification that binds x. Note that all occurrences of the same variable x share these two

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

places. The two transitions implements the reading of the current value of x.

Second, consider a negation $\neg P$, see Figure 3, box C. The transition *call_P* transfers the "control" to the subnet for P. The two other transitions implement the negation.

Third, consider a conjunction $P \wedge Q$, see Figure 3, box D. The transition *call_P* transfers the "control" to the subnet for P. The four other transitions implement the conjunction.

Fourth, consider an existential quantification $\exists x.P$, see Figure 3, box E. The places *x_is_T* ("x is true") and *x_is_F* ("x is false") are the ones we mentioned above. The transition *call_P_with_x_T* assigns true to x and transfers the "control" to the subnet for P. In case P was not true, the transition *call_P_with_x_F* assigns false to x and transfers again the "control" to the subnet for P.

If a formula P is open, then we can obtain an *extended* net for P as follows. For every free variable x in P we extend the net with two places *x_is_T* and *x_is_F* and mark exactly one of them. This marking may be thought of as assigning a value to x.

The following fact expresses a relation between each formula P and the extended net for P. The proof is by straightforward induction on the structure of P.

- **Fact** Let P be a formula generated from the above grammar and consider the extended net for P. In the following we discount the marking of the places for free variables; the marking of these are invariant. From the marking $\{in\_P\}$, eventually either $\{P\_T\}$ or $\{P\_F\}$ will be reached. The former is reached if and only if P is true under the given assignment of its free variables, and the latter if not.

Using this observation it is easy to see that the marking $\{G\_T\}$ is reachable in the net for G if and only if G is true.

Clearly, the net for G is 1-safe. Notice that for each reachable marking at most one transition is enabled.
□

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

Proof. To show that the deadlock problem is in PSPACE, given a 1-safe net N guess a marking M of N, and check in constant space if it is a deadlock; guess an occurrence sequence from the initial marking (only the marking reached so far needs to be stored, which uses linear space); check after each step if the occurrence sequence constructed so far leads to M.

To prove completeness, we reduce the problem QUANTIFIED BOOLEAN FORMULAS to the deadlock problem. Extend the net in the proof of Theorem 5 with the transition

$$G\_F \rightarrow G\_F$$

Clearly, the new net has a deadlock if and only if $\mathcal{F}$ is true.
□

The deadlock and and reachability problems turn out to be PSPACEcomplete even for 1-conservative unary 1-safe nets. This follows directly from the

constructions in the proof of Theorem 5 and the following "conservativeness" observation.

First, we define the notion of reachability graph. The reachability graph of a net N is the edge-labeled graph whose vertices are the reachable markings of N; if $M \xrightarrow{t} M'$ for a reachable marking M, then there is an edge from M to M' labeled with t.

Fact 7 *There is a linear time algorithm which converts a* 1 *-safe net N into a* 1 *-conservative* 1 *-safe net N' with the following property: there exists a simple function f from the markings of N to the markings of N' such that (1) M is reachable in N iff f(M) is reachable in N'; (2) the initial marking of N is mapped by f to the initial marking of N'; and (3) M is a deadlock of N iff f(M) is a deadlock of N'. Hence the conotruction 'preserves' reachability and the existence of deadlocks.*

For $N = (P, T, F, M_0)$, the net N' is constructed by adding for every place p of P a new place $\bar{p}$ called the complement of p. Then, for every arc (p, t) of $F \backslash F^{-1}$, a new arc $(t, \bar{p})$ is added; similarly, for every arc (t, p) of $F \backslash F^{-1}$, a new arc $(\bar{p}, t)$ is added. Finally M_0' is defined by $M_0'(p) = M_0(p)$ for every place p of N, and $M_0'(\bar{p}) = 1 - M_0(p)$ for each complement place. The construction is very similar to the one of [33], and therefore we omit the proof of the result; the only difference is the special treatment of the case in which two arcs (p, t) and (t, p) exist.

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 synchronizations. In particular, in 1-safe free-choice net the phenomenon known as *confusion* [36] is ruled out.

5.1 Acyclic nets

A net $N = (P, T, F, M_0)$ is said to be acyclic if F^+ (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 (assuming NP *neq* PSPACE).

> **Theorem 8** *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 [35]. 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 can compare these results with the ones corresponding to the general case.

> **Theorem 9** *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, [30]). Since INTEGER LINEAR PROGRAMMING is in NP [20], 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 confiicts are structurally ruled out (actually, this depends slightly on the notion of confict 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 [21][22] that the reachability, liveness, and deadlock problems for 1-safe conflict-free nets are solvable in polynomial time. They also show that, for Place/Transition nets, the deadlock and liveness problems are still polynomial, whereas the reachability problem becomes NP-complete [21][22].

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. The interested reader can find a formal definition in [24][19]. Figure 3 shows a non-free-choice net (on the left), and its released form (on the right).

Perhaps surprisingly, the liveness problem is polynomial for this class.

Theorem 10 *The liveness problem for free-choice* 1*-safe nets is solvable in polynomial time.*

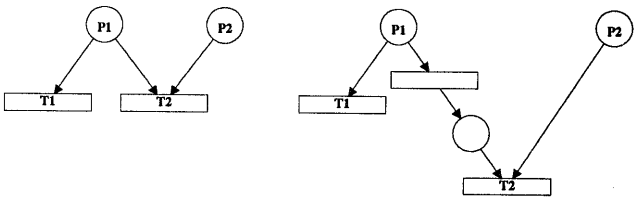

Figure 4: A net and its released form.

Proof. See the paper by Esparza and Silva [12], and the paper 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.

Let N be a net and Q a set of places of N. For a marking M of N, $M(Q)$ denotes the total number of tokens that M puts in the places of Q (formally, $M(Q) = \sum_{p \in Q} M(p)$. The set Q is said to be marked at M if $M(Q) > 0$, and unmarked at M if $M(Q) = 0$.

A subset Q of places of N is a *siphon* if $^\bullet Q \subseteq Q^\bullet$, and a *trap* if $Q^\bullet \subseteq^\bullet Q$.

We use some well known lemmata about siphons and traps. They can all be found in ref17 or - a more accessible reference - in [2].

Lemma 11 *Let N be a net, and M a marking of N.*

1. *If Q is a siphon of N unmarked at M, then Q remains unmarked at all markings reachable from M.*

2. *If Q is a trap of N marked at M, then Q remains marked at all markings reachable from M.*

Proof. Follows easily from the definitions of siphon, trap, and the occurrence rule.
□

Lemma 12 *Let M be a deadlock of a net N. Then, the set of places of N unmarked at M is a siphon of N.*

Proof. Let Q be the set of places of N unmarked at M. It suffices to observe that, since M is a deadlock, every transition has some place in its preset which is unmarked at M. So $Q^\bullet$ contains all the transitions of N and, since ${}^\bullet Q$ is a subset of them, Q is a siphon.
□

Lemma 13 *Let N be a free-choice net with initial marking M_0. Let Q be a siphon of N which contains no trap marked at M_0. Then, there exists a reachable marking M such that Q is unmarked at it.*

Proof. See [17][2]. This result is part of the proof of Commoner's theorem.
□

Using these lemmata, we can now characterise when a free-choice net has a deadlock.

Lemma 14 *Let N be a free-choice net. N has a deadlock iff there exists a siphon Q of N such that:*

1. *for every transition t of N, Q contains some place of ${}^\bullet t$, and*
2. *Q contains no trap marked at the initial marking.*

Proof. ($\Rightarrow$): Let M be a deadlock of N. Define Q as the set of places of N unmarked at M. By Lemma 12, Q is a siphon. Since no transition of N is enabled at M, we have that, for every transition t, Q contains some place of ${}^\bullet t$.

To prove (2), assume that Q contains a trap marked at the initial marking. Then, since marked traps remain marked by Lemma 11, this trap is marked at M. So Q is marked at M too, which contradicts the definition of Q.

($\Leftarrow$): By Lemma 13, there exists a reachable marking M such that $M(Q) = 0$. Since Q contains some place of the preset of each transition, no transition is enabled at M. So M is a deadlock.
□

Theorem 15 *The deadlock problem for* 1*-safe free-choice nets is NP-complete.*

Proof. To solve the problem in nondeterministic polynomial time, we use Lemma 14. Guess for each transition t of the net a place of ${}^\bullet t$. Check in polynomial time if the guessed set of places is a siphon; then, check in polynomial time that it contains no trap marked at the initial marking using Starke's algorithm to find the maximal trap contained in a given siphon [34] (see [9] for a reference in English).

We prove completeness by reducing the satisfiability problem of propositional formulas in conjunctive normal form (CON-SAT) to the deadlock problem.

An instance ϕ of CON-SAT is a conjunction of clauses $C_1, \ldots, C_m$ over variables $x_1, \ldots, x - n$. A clause is a disjunction of literals. A literal l_i is either a variable x_i or its negation $\bar{x_i}$.

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 [24] to prove the NP-completeness of liveness in general free-choice nets. We describe the set P of places and the set T of transitions of N, together with their presets and postsets. The set P contains the following elements:

(a) for every $1 \leq i \leq n$, places $A_i, x_i, \bar{x_i}$

(b) for each clause C_j and every literal l_i appearing in C_j, a place (l_i, C_j) and

(c) for each clause C_j, a place F_j.

The transitions in T are defined as follows:

1. for each literal $l_i, A_i \rightarrow l_i$,
2. for each literal $l_i, l_i \rightarrow$
$$\sum_{\bar{l_i} \in C_j} (l_i, C_j)$$
3. for each clause C_j,
$$\sum_{l_i \in C_j} (l_i, C_j) \rightarrow F_j$$
and
4. for each clause $C_j, F_j \rightarrow F_j$.

The marking M_0 is the set $\{A_i \mid 1 \leq i \leq n\}$.

An occurrence sequence of N is a *truth sequence* if:

- for every variable x_i, it contains one of the two transitions $A_i \rightarrow x_i, A_i \rightarrow \bar{x_i}$, and
- it only enables transitions of type (3), if any.

A truth sequence σ is associated to the assignment $f : \{x_1, \ldots, x_n\} \rightarrow \{true, false\}$ given by $f(x_i) = true$ iff the transition $A_i \rightarrow x_i$ occurs in σ. The following fact follows easily from the construction of N:

- Fact The marking reached by a truth sequence enables a type (3) transition iff the corresponding clause C_j is false under f.

Assume ϕ is satisfiable. Then, there exists an assignment f which makes all clauses true. By the fact above, any truth sequence associated to f leads to a deadlock.

Now, assume that M is a deadlock of N. It follows from the construction that M only marks places of the form (l_i, C_j), and that any occurrence sequence that leads to M is a truth sequence. By the fact above, no clause is false under the assignment associated to σ. So ϕ is satisfiable.
□

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 [24]. Finally, our proof of membership in NP for the deadlock problem did not rely on 1-safeness; therefore, the deadlock problem is also NP-complete for Place/Transition free-choice nets.

6 Other Problems

There exist other problems concerning Petri nets which have received attention in the literature.

The *containment problem* for two nets with the same set of places is the problem of deciding whether all reachable markings of the first are reachable in the second.

Given two 1-safe markings M, M' of a net, M is *covered* by M' if $M \subseteq M'$. The *coverability problem* for a given net N and a marking M of N is the problem of deciding whether some reachable marking of N covers M.

A net N is said to be *persistent* [25] if for every reachable marking M, if two different transitions t, t' are enabled at M then $M \xrightarrow{t} M' \xrightarrow{t} M''$ for some markings M', M''. The *persistency problem* for a net is the problem of deciding whether the net is persistent. Notice that unary nets are persistent.

Let $N = (P, T, F, M_0)$ be a net. For any subset T_0 of T let h_{T_0} be the "erasing" homomorphism from T^* to T_0^* which erases elements from $T \setminus T_0$.

For a transition $t \in T \setminus T_0$ we say that T_0 controls t by an occurrence sequence γ in T_0^* if for every occurrence sequence σ from M_0 if $h_{T_0}(\sigma) = \gamma$ then t is not enabled at the marking M reached by the occurrence of σ. Crudely speaking, once γ has occurred, even interleaved with transitions of $T \setminus T_0$, t cannot occur until some transition of T_0 occurs. T_0 is said to control t if T_0 can control t by at least one sequence γ. The *controllabilty problem* [24] for a net is the problem of deciding whether T_0 controls t given N, T_0, and t as above.

For arbitrary Petri nets, the containment problem is undecidable [19], whereas the coverability, persistency and controllability problems are EXPSPACE-hard. It is shown by Howell and Rosier in [21][22] that the coverability problem for 1-safe confict-free nets is solvable in polynomial time.

We study the first three of these problems in the 1-safe case.

Theorem 16 *The containment, coverability and persistency problems for* 1*-safe nets are PSPACE-complete.*

Proof. We show that each of the three problems is in PSPACE. First, consider the containment problem. Given two nets, guess a marking and check in linear space that the marking is reachable in the first net and unreachable in the second net. This shows that the containment problem is in co-NPSPACE and thus in PSPACE (by Savitch's theorem and because space complexity classes are closed under complementation).

Second, consider the coverability problem. Given a 1-safe net N and a marking M of N, guess both a marking $M' \supseteq M$ and, step by step, an occurrence sequence from the initial marking (only the marking reached so far needs to be stored, which uses linear space); check after each step if the occurrence sequence constructed so far leads to M'.

Third, consider the persistency problem. Proceed as above, this time guessing a marking M of N that enables two different transitions t and t'. If M is reachable, then check in linear space that t' cannot occur after the occurrence of t.

To prove that each of the three problems is PSPACE-hard, we use the same construction as in the proof of PSPACE-hardness of reachability. For each of the following arguments, suppose we are given a quantified boolean formula

$\mathcal{F}$. To begin with, transform it into an equivalent formula G as was done for Theorem 5.

First, consider the containment problem. Construct both the same 1-safe net N as in the proof of Theorem 5 and the following net N'. The net N' is obtained from N by removing all transitions and taking $\{G\_T\}$ as initial marking. For convenience we construct a net whose places have empty presets and postsets (isolated nodes), see remark at the beginning of section 2. The PSPACE-hardness can be shown for nets satisfying the assumptions of no isolated nodes. Clearly, the set of reachable markings of N' is $\{G\_T\}$, and therefore it is contained in the set of reachable markings of N if and only if $\mathcal{F}$ is true.

Second, consider the coverability problem. Clearly, there is a reachable marking in N that covers $\{G\_T\}$ if and only if $\mathcal{F}$ is true.

Third, consider the persistency problem. Extend the net in the proof of Theorem 5 with two new places $\{V, W\}$ and the transitions

$$\begin{array}{lcl} G\_F & \rightarrow & V \\ G\_F & \rightarrow & W \end{array}$$

Clearly, the new net is persistent if and only if $\mathcal{F}$ is true.
□

The proof of the result on controllability [24], [Theorem 4.1] was in fact given for 1-conservative free-choice nets, and also works when restricted to 1-safe nets. This is the only one of the problems we consider for which the complexity does not decrease for 1-safe nets.

Using the techniques from the proofs of Theorem 5 and 16 one can proceed to prove that numerous other problems for 1-safe nets are PSPACE-complete: "is there an infinite occurrence sequence?", "can a certain transition ever occur?", "is a certain transition live?", etc. The interested reader will find no problem in carrying out the corresponding proofs.

7 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 [10] 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 *Aduances in Petri Nets 1992*, pages 304-351. Springer-Verlag (LNCS 609), 1992.

[2] Eike Best and Jörg Desel.
Partial order behaviour and structure of Petri nets.
Formal Aspects of Computing, 2:123-138, 1990.

[3] 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 6O9), 1992.

[4] Eike Best and C. Fernández.
Nonsequential Processes - a Petri Net View.
EATCS Monographs on Theoretical Computer Science Vol.13, 1988.

[5] 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.

[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 Net 1992*, pages 304-351. Springer-Verlag (*LNCS* 609), 1992.

[9] Jörg Desel and Javier Esparza.
Reachability in reversible free choice systems.
In *Proc. STACS'91*, pages 384-397. Springer-Verlag (LNCS 480), 1991.

[10] 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.

[11] Javier Esparza.
Model checking using net unfoldings.
In *Proc. TAP-SOFT'98*, pages 613-628. Springer-Verlag (LNCS 668), 1993.

[12] Javier Esparza and Manuel Silva.
A polynomial-time algorithm to decide liveness of bounded free choice nets.
Theoretical Computer Science, 102:185-205, 1992.

[13] Michael R. Garey and David S. Johnson.
Computers and Intractability.
Freeman, 1979.

[14] Hartmann J. Genrich and Kurt Lautenbach.
Synchronisationsgraphen.
Acta Informatica, 2:143-161, 1973.

[15] 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.

[16] 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.

[17] Michel Hack.
Analysis of production schemata by Petri nets.
Master's thesis, MIT, 1972.

[18] 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.

[19] Michel Hack.
The equality problem for vector addition systems is undecidable.
Theoretical Computer Science, 2:77-95, 1976.

[20] John E. Hopcroft and Jeffrey D. Ullman.
Introduction to Automata Theory, Languages and Computation.
Addison-Wesley Publishing Company, 1979.

[21] 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.

[22] 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.

[23] Lalita Jategaonkar and Albert Meyer.
Deciding true concurrency equivalences on finite safe nets.
In *Proc. ICALP'93*, pages 519-531, 1993.

[24] Neil D. Jones, Lawrence H. Landweber, and Y. Edmund Lien.
Complexity of some problems in Petri nets.
Theoretical Computer Science, 4:277-299, 1977.

[25] Lawrence H. Landweber and E. L. Robertson.
Properties of conflictfree and persistent Petri nets.
Journal of the ACM, 3:352-364, 1975.

[26] Richard J. Lipton.
The reachability problem requires exponential space.
Technical Report 62, Yale University, 1976.

[27] Ernst W. Mayr.
An algorithm for the general Petri net reachability problem.
SIAM Journal on Computing, 13:441-460, 1984.

[28] 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.

[29] José Meseguer and Ugo Montanari.
Petri nets are monoids.
Information and Computation, 88:105-155, 1990.

[30] Tadao Murata.
Petri nets: Properties, analysis and applications.
In *Proc. of the IEEE*, volume 77(4), pages 541-580, 1989.

[31] Mogens Nielsen, Grzegorz Rozenberg, and P.S. Thiagarajan.
Behavioural notions for elementary net systems.
Distributed Computing, 4(1):45-57, 1990.

[32] Ernst R. Olderog.
Nets, Terms and Formulas.
Cambridge University Press, 1991. *Number 23 Tracts in Theoretical Computer Science.*

[33] Wolfgang Reisig.
Petri Nets - An Introduction.
EATCS Monographs in Computer Science Vol.4, 1985.

[34] Peter H. Starke.
Analyse von Petri-Netz-Modellen.
Teubner, 1990.

[35] Iain A. Stewart.
On the reachability problem for some classes of Petri nets.
Research Report, University of Newcastle upon Tyne, 1992.

[36] P.S. Thiagarajan.
Elementary net systems.
In Advances in Petri Nets 1986, part I, pages 26-59. Springer-Verlag (lNCS 254), 1987.

[37] 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.

[38] Glynn Winskel.
Petri nets, algebras, morphisms and compositionality.
Information and Computation, 72(3):197-238, 1987.

[39] 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.