Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 59
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 59
Espoo 2000 HUT-TCS-A59

COMPUTATIONAL COMPLEXITY OF THE PLACE/TRANSITION-NET SYMMETRY REDUCTION METHOD

Tommi Junttila

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 59
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 59
Espoo 2000 HUT-TCS-A59

COMPUTATIONAL COMPLEXITY OF THE PLACE/TRANSITION-NET SYMMETRY REDUCTION METHOD

Tommi Junttila

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory for Theoretical Computer Science

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietojenkäsittelyteorian laboratorio

Distribution:
Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O.Box 5400
FIN-02015 HUT
Tel. +358-0-451 1
Fax. +358-0-451 3369
E-mail: lab@tcs.hut.fi

ISBN 951-22-4984-7
ISSN 0783-5396

Picaset Oy
Helsinki 2000

ABSTRACT: Computational complexity of the sub-tasks appearing in the symmetry reduction method for Place/Transition-nets is studied. The first task of finding the automorphisms (symmetries) of a net is shown to be polynomial time many-one equivalent to the problem of finding the automorphisms of a graph. The problem of deciding whether two markings are symmetric is shown to be equivalent to the graph isomorphism problem. Surprisingly, this remains to be the case even if the generators for the automorphism group of the net are known. The problem of constructing the lexicographically greatest marking symmetric to a given marking (a canonical representative for the marking) is classified to belong to the lower levels of the polynomial hierarchy, namely to somewhere between $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}^{\mathbf{NP}}$. It is also discussed how the self-symmetries of a marking can be exploited. Calculation of such symmetries is classified to be as hard as computing graph automorphism groups. Furthermore, the coverability version of testing marking symmetricity is shown to be an **NP**-complete problem. It is shown that unfortunately canonical representative markings and the symmetric coverability problem cannot be combined in a straightforward way.

KEYWORDS: Petri nets, symmetry

CONTENTS

1 INTRODUCTION

Symmetries in a Petri net yield symmetries in its behaviour. This symmetry can be exploited to alleviate the state-space explosion problem occurring in the reachability analysis of nets. The symmetry reduction method was introduced by Huber et al. [1985; 1991] for colored high-level Petri nets. The method was applied to low-level nets, the formalism of this paper, by Starke [1991] and further studied in [Schmidt and Starke 1991; Schmidt 1997; 1999; 2000a; 2000b]. The main idea of the method is that the symmetries (automorphisms) of a low-level net produce corresponding symmetries to the state-space of the net. For many verification tasks, such as deadlock checking, it is sufficient to inspect only one marking in each set of mutually symmetric markings (orbit). Thus a (potentially exponentially smaller) quotient reachability graph can be constructed instead of the normal reachability graph. Schmidt and Starke have presented algorithms for solving many of the problems involved in the method [Schmidt and Starke 1991; Schmidt 1997; 1999; 2000a; 2000b]. However, the topic of this paper, the computational complexity issues of the sub-tasks appearing in the method, has not been addressed before.[1]

The problem of finding the automorphisms of a net is easily proven to be as hard as finding the automorphisms of a graph. This is not surprising since nets can be seen as labelled directed graphs. We show that the problem of deciding whether two markings are symmetric is equivalent (in the polynomial time many-one reduction sense) to the graph isomorphism problem. Interestingly, this remains to be the case even if the automorphism group of the net is known. To avoid the pair-wise comparison of markings for symmetry during the quotient reachability graph generation, a canonical representative marking for the whole orbit of markings can be generated. This problem is of course at least as hard as the graph isomorphism problem since solving it solves the marking symmetry problem, too. In this paper we show that computing the most obvious canonical representative marking, namely the lexicographically greatest marking in the orbit, is a problem whose complexity is somewhere between $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}^{\mathbf{NP}}$.

We also introduce the concept of marking-stabilizers (self-symmetries of markings) which are symmetries of the net that map a marking to itself. We show that computing the marking-stabilizer group for a marking is as hard as computing the automorphism group of a graph. We show how marking-stabilizers improve the generation of quotient reachability graphs by allowing us to ignore some symmetric transitions. We also demonstrate how marking-stabilizers can speed up the "loop over all symmetries"-approach for marking symmetry.

As the last problem we consider the coverability problem under symmetries. It asks, given two markings of a net, whether there is a net automorphism such that the first marking covers the second marking when permuted with the automorphism. An interesting phenomenon happens here: the problem becomes **NP**-complete instead of staying as hard as graph isomorphism. Furthermore, we show that the symmetric coverability problem does

[1] For some complexity theoretical results concerning a high-level Petri net formalism, see [Junttila 1999].

not, unfortunately, allow itself to be integrated into the canonical representative marking approach in a straightforward way.

The paper is structured as follows. Section 2 gives the necessary preliminaries and Sec. 3 defines P/T-nets and their symmetries. The complexities of the fundamental problems of (i) computing net automorphism groups, (ii) deciding whether two markings are symmetric and (iii) the construction of canonical representative markings are proven and discussed in Sec. 4. Section 5 presents the concept, use and computational complexity of marking-stabilizers while Sec. 6 deals with the symmetric coverability problem. Finally, Sec. 7 concludes the paper.

2 PRELIMINARIES

2.1 Computational Complexity Theory

For computational complexity theory in general, see e.g. [Garey and Johnson 1979; Papadimitriou 1995]. Letting $A, B \subseteq \Sigma^*$ be languages (decision problems) over some finite alphabet Σ, we say that A *polynomial time many-one reduces* to B, denoted by $A \leq^{\mathrm{p}}_{\mathrm{m}} B$, if there is a polynomial time computable function $R : \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$ it holds that $x \in A \Leftrightarrow R(x) \in B$. If both $A \leq^{\mathrm{p}}_{\mathrm{m}} B$ and $B \leq^{\mathrm{p}}_{\mathrm{m}} A$ hold, we say that A and B are *polynomial time many-one equivalent*.In this paper we omit the prefix "polynomial time" and simply say that A many-one reduces to B or that A and B are many-one equivalent.

For function problems $f, g : \Sigma^* \to \Sigma^*$, we say that f *polynomial time many-one reduces* to g, denoted by $f \leq^{\mathrm{p}}_{\mathrm{m}} g$, if there are polynomial time computable functions $R, S : \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$ it holds that $f(x) = S(g(R(x)))$. Our reductions are similar to the *metric reductions* in [Krentel 1988] as long as we are dealing with complexity classes above and including **P**. On the other hand, our reductions may be a bit stronger than those in [Papadimitriou 1995] since we use polynomial time instead of logarithmic space. Many-one equivalence for function problems is defined in the same way as for decision problems.

The usual complexity classes of problems decidable in polynomial time with deterministic and non-deterministic Turing machines are denoted by **P** and **NP**, respectively. **FP** (**FNP**) means the class of function problems computable by (non-)deterministic Turing machines in polynomial time. $\mathbf{FP}^{\mathbf{NP}}$ ($\mathbf{FP}^{\mathbf{NP}[\log n]}$) is the class of function problems computable in polynomial time by deterministic Turing machines that can access an **NP**-oracle polynomially (logarithmically) many times w.r.t. the input size.

2.2 Graph-Theoretical Problems

Since nets can be seen as directed labelled graphs and graph theory is a well-studied field, we use graph theoretical problems to classify the problems concerning net symmetries.

Definition 2.1 A labelled directed graph *is a triple* $G = \langle V, E, L \rangle$ *where* V *is a finite set of* vertices, $E \subseteq V \times V$ *is the set of* edges *and the function* L

assigns each vertex and each edge a label.

A labelled directed graph is *undirected* if its vertex set is anti-reflexive and symmetric. Furthermore, it is *non-labelled* if the range of the labeling function is a unit set (all labels are the same). A non-labelled undirected graph is called simply a *graph*. Two labelled directed graphs, $G_1 = \langle V_1, E_1, L_1 \rangle$ and $G_2 = \langle V_2, E_2, L_2 \rangle$, are *isomorphic* iff there is a bijective mapping (isomorphism) $\pi : V_1 \to V_2$ such that (i) $\langle v_1, v_2 \rangle \in E_1$ iff $\langle \pi(v_1), \pi(v_2) \rangle \in E_2$, (ii) $L_2(\pi(v)) = L_1(v)$ for all $v \in V$ and (iii) $L_2(\langle \pi(v_1), \pi(v_2) \rangle) = L_1(\langle v_1, v_2 \rangle)$ for all $\langle v_1, v_2 \rangle \in E_1$.

Problem 2.2 GRAPH ISOMORPHISM. *Given two labelled directed graphs, are they isomorphic?*

It is easy to see, based on results by Miller [1979], that the graph isomorphism problems for (non-labelled, undirected) graphs and labelled directed graphs are many-one equivalent and therefore we do not distinguish between them in this work. The graph isomorphism problem is an interesting problem because, although it clearly belongs to **NP**, it has not been shown to belong to **P** nor to be **NP**-complete but is one of the main candidates for a problem to be in between (such problems must exist if $\mathbf{P} \neq \mathbf{NP}$ as is widely believed). For more information about the computational complexity of the graph isomorphism problem, the reader is referred to [Köbler et al. 1993].

A concept closely related to graph isomorphism is that of graph automorphisms. An *automorphism* π of a labelled directed graph $G = \langle V, E, L \rangle$ is an isomorphism from G to itself. The *set of all automorphisms* of G is denoted by $\mathrm{Aut}(G)$.

Problem 2.3 GRAPH AUTOMORPHISMS. *Given a graph G, find* $\mathrm{Aut}(G)$.

Again, the complexity of the graph automorphism problem is the same for graphs and labelled directed graphs. GRAPH AUTOMORPHISMS is a function problem that is *polynomial time equivalent* to GRAPH ISOMORPHISM, that is, if either has a polynomial time algorithm, then (and only then) both have.

For a finite set A, the set of all bijections (permutations) on A is denoted by $\mathrm{Sym}(A)$ and is a group under the function composition operation $\circ$. Obviously, $\mathrm{Aut}(G)$ for a labelled directed graph $G = \langle V, E, L \rangle$ is a sub-group of $\mathrm{Sym}(V)$. In this paper it is assumed that permutation groups (sub-groups of $\mathrm{Sym}(A)$ for a set A) are given by means of their generator sets. We then know that we can construct, in polynomial time w.r.t. the size of the permuted set and the number of generators, a normal form representation of the group. Furthermore, we can test in polynomial time whether a permutation belongs to the group [Furst et al. 1980]. For permutation group algorithms, see e.g. [Butler 1991; Kreher and Stinson 1999].

3 SYMMETRIES OF PLACE/TRANSITION-NETS

The presentation in this section is based on [Starke 1991; Schmidt and Starke 1991; Schmidt 1997; 2000a].

3.1 P/T-Nets

A *Place/Transition-net* (or a P/T-net) is a tuple $N = \langle P, T, F, V, M_0 \rangle$, where

1. P is a finite, non-empty set of *places*,
2. T is a finite, non-empty set of *transitions* such that $P \cap T = \emptyset$,
3. $F \subseteq (P \times T) \cup (T \times P)$ is the *flow-relation* (also called the *set of arcs*),
4. $V : F \to \mathbb{N}_+$ maps each arc in F with a multiplicity (we define that $V(\langle x, y \rangle) = 0$ if $\langle x, y \rangle \notin F$) and
5. $M_0 : P \to \mathbb{N}$ is the initial marking of N.

A *marking* of N is a function $M : P \to \mathbb{N}$ and the *set of all markings* of N is denoted by $\mathbb{M}$. A marking M can also be denoted by the formal sum $\sum_{p \in P} M(p)p$. For two markings, M and M', $M \leq M'$ iff $(\forall p \in P)(M(p) \leq M'(p))$. A transition $t \in T$ is *enabled* in a marking M if $V(\langle p, t \rangle) \leq M(p)$ for all $p \in P$. If t is enabled in M, it may *fire* and transform M into M' defined by $M'(p) = M(p) - V(\langle p, t \rangle) + V(\langle t, p \rangle)$ for all $p \in P$. This is denoted by $M\ [t\rangle\ M'$. The *reachability graph* of N is the labelled transition system $\mathrm{RG}(N) = \langle Q, \Delta, M_0 \rangle$, where $Q \subseteq \mathbb{M}$ and $\Delta \subseteq Q \times T \times Q$ are defined inductively by:

1. $M_0 \in Q$;
2. if $M \in Q$ and $M\ [t\rangle\ M_1$, then $M_1 \in Q$ and $\langle M, t, M_1 \rangle \in \Delta$; and
3. nothing else is in Q or Δ.

A marking M is *reachable* if it belongs to Q.

3.2 Symmetries of P/T-nets

Symmetries of the net N are automorphisms of the net when seen as labelled directed graph, i.e., permutations that respect node type, flow relation and arc annotations.

Definition 3.1 *A* symmetry *(an automorphism) of N is a permutation $\sigma \in \mathrm{Sym}(P \cup T)$ which*

1. *respects node type, i.e., $\sigma(P) = P$ and $\sigma(T) = T$;*
2. *respects the flow relation: $\langle x, y \rangle \in F \Leftrightarrow \langle \sigma(x), \sigma(y) \rangle \in F$ for all $x, y \in P \cup T$; and*
3. *respects the arc multiplicities: $V(\langle x, y \rangle) = V(\langle \sigma(x), \sigma(y) \rangle)$ for all $\langle x, y \rangle \in F$.*

The set of all symmetries *of N (the* automorphism group *of N) is denoted by $\mathrm{Aut}(N)$ and is a sub-group of $\mathrm{Sym}(P \cup T)$.*

A symmetry σ of N is extended to operate on the markings of N by letting the marking $\sigma(M)$ be the one satisfying $(\sigma(M))(p) = M(\sigma^{-1}(p))$, or equivalently, $(\sigma(M))(\sigma(p)) = M(p)$. We say that two markings, M and M', of N are *symmetric*, denoted by $M \equiv M'$, if $(\exists \sigma \in \mathrm{Aut}(N))(\sigma(M) = M')$. The set of markings symmetric to a marking M is the equivalence class denoted by $[M]$ (the *orbit* of M). It is these equivalence classes that are exploited in the symmetry reduction method. Formally, a *quotient reachability graph* of N is a labelled transition system $\langle \tilde{Q}, \tilde{\Delta}, M_0' \rangle$, where $M_0' \in [M_0]$ and $\tilde{Q} \subseteq \mathbb{M}$, $\tilde{\Delta} \subseteq \tilde{Q} \times T \times \tilde{Q}$ are defined inductively by:

1. $M_0' \in \tilde{Q}$;
2. if $M \in \tilde{Q}$ and $M\ [t\rangle\ M_1$, then $M_1' \in \tilde{Q}$ and $\langle M, t, M_1' \rangle \in \tilde{\Delta}$ for a $M_1' \in [M_1]$; and
3. nothing else is in $\tilde{Q}$ or $\tilde{\Delta}$.

Various properties, such as deadlock freedom, of the net N can be checked by using a quotient reachability graph of N. For more on these properties and temporal logic model checking under symmetries, see e.g. [Starke 1991; Jensen 1995; 1996; Clarke et al. 1996; Emerson and Sistla 1996; Gyuris and Sistla 1999].

The *integration problem* in the (inductive) generation of quotient reachability graphs is [Schmidt 1999; 2000b]:

Problem 3.2 *Given a set $\tilde{Q}$ of already visited markings and a newly generated marking M, find out whether there is a marking $M' \in \tilde{Q}$ such that $M \equiv M'$.*

There are three basic ways to solve the integration problem [Schmidt 1999; 2000b]:

1. When $\mathrm{Aut}(N)$ is known, loop over all symmetries in it and for each σ of them, check whether $\sigma(M) \in \tilde{Q}$. Of course, for $\mathrm{Aut}(N)$ with large order this is highly infeasible.
2. For each marking $M' \in \tilde{Q}$, check whether $M' \equiv M$. Symmetry respecting hash functions [Schmidt 1999; 2000a; 2000b] can be used to prune the set of markings of $\tilde{Q}$ that need to be checked.
3. Build a canonical representative marking for M and check whether it is in $\tilde{Q}$.

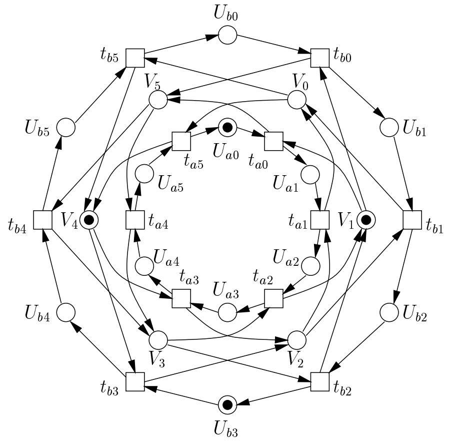
Example 3.3 Consider the variant of Genrich's railroad system net [Genrich 1991] shown in Fig. 1(a). Its reachability graph is shown in Fig. 1(b). The group $\mathrm{Aut}(N)$ is generated by the rotation

$$\sigma_{\mathrm{rot}} = \left(\begin{smallmatrix} U_{a0} & U_{a1} & U_{a2} & U_{a3} & U_{a4} & U_{a5} & U_{b0} & \cdots & U_{b5} & V_0 & \cdots & V_5 & t_{a0} & \cdots & t_{a5} & t_{b0} & \cdots & t_{b5} \\ U_{a1} & U_{a2} & U_{a3} & U_{a4} & U_{a5} & U_{a0} & U_{b1} & \cdots & U_{b0} & V_1 & \cdots & V_0 & t_{a1} & \cdots & t_{a0} & t_{b1} & \cdots & t_{b0} \end{smallmatrix}\right)$$

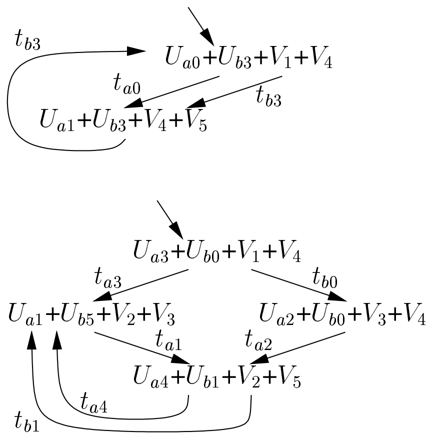
and the swapping of train identities

$$\sigma_{\mathrm{swap}} = \left(\begin{smallmatrix} U_{a0} & \cdots & U_{a5} & U_{b0} & \cdots & U_{b5} & V_0 & \cdots & V_5 & t_{a0} & \cdots & t_{a5} & t_{b0} & \cdots & t_{b5} \\ U_{b0} & \cdots & U_{b5} & U_{a0} & \cdots & U_{a5} & V_0 & \cdots & V_5 & t_{b0} & \cdots & t_{b5} & t_{a0} & \cdots & t_{a5} \end{smallmatrix}\right).$$

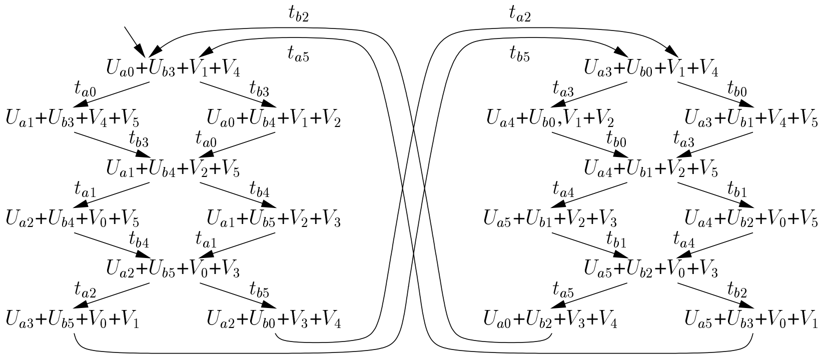
Now the initial marking $M_0 = U_{a0}+U_{b3}+V_1+V_4$ is symmetric to the marking $M = U_{a4}+U_{b1}+V_2+V_5$ as $\sigma_{\text{swap}}(\sigma_{\text{rot}}(M_0)) = \sigma_{\text{swap}}(U_{a1}+U_{b4}+V_2+V_5) = M$. The orbit of M_0 consists of markings M_0, $U_{a1} + U_{b4} + V_2 + V_5$, $U_{a2} + U_{b5} + V_0 + V_3$, $U_{a3} + U_{b0} + V_1 + V_4$, $U_{a4} + U_{b1} + V_2 + V_5$ and $U_{a5} + U_{b2} + V_0 + V_3$. Figure 1(c) shows two quotient reachability graphs of the net where the upper one is minimal in the sense that it contains only one marking per orbit. ♣



(a) The net.



(c) Two quotient reachability graphs.



(b) The reachability graph.

Figure 1: A net for a railroad system.

4 COMPLEXITY OF SUB-PROBLEMS

4.1 Computing Net Automorphisms

The first problem is to find the automorphism group of a net.

Problem 4.1 NET AUTOMORPHISMS. *Given a net* N, *compute* $\text{Aut}(N)$.

Since nets are directed labelled graphs, it is easy to show that NET AUTOMORPHISMS is equivalent to the GRAPH AUTOMORPHISMS problem.

Theorem 4.2 NET AUTOMORPHISMS *is many-one equivalent to* GRAPH AUTOMORPHISMS.

Proof. We first reduce from GRAPH AUTOMORPHISM to NET AUTOMORPHISMS. Given a directed graph $G = \langle V, E \rangle$, we construct the net $N = \langle P, T, F, V, M_0 \rangle$ such that $P = V$, $T = E$, $F = \{\langle v, \langle v, v' \rangle \rangle \mid \langle v, v' \rangle \in E\} \cup \{\langle \langle v, v' \rangle, v' \rangle \mid \langle v, v' \rangle \in E\}$ and $V(f) = 1$ for all $f \in F$. The initial marking is irrelevant. It follows directly from the definitions that the group $\mathrm{Aut}(N)$ restricted to the set P of places is $\mathrm{Aut}(G)$. To reduce the other way round, just interpret the net as a directed labelled graph. Edges are labelled with the corresponding multiplicities while the nodes corresponding to places are labelled with "P" and those to transitions with "T", for instance, to separate them. Clearly the automorphism group the graph is the automorphism group of the net. □

4.2 Testing Marking Symmetricity

Let us next consider the problem of deciding whether two markings of a net N are symmetric. We consider two cases: the one in which the automorphism group of N is not known and the other in which it is.

Problem 4.3 UNINFORMED MARKING SYMMETRY (UMS). *Given a net N and two markings of N, are the markings symmetric?*

Problem 4.4 INFORMED MARKING SYMMETRY (IMS). *Given a net N, the group $\mathrm{Aut}(N)$ and two markings of N, are the markings symmetric?*

Clearly IMS $\leq^{\mathrm{p}}_{\mathrm{m}}$ UMS. We now show in two parts that both IMS and UMS are many-one equivalent to GRAPH ISOMORPHISM.

Lemma 4.5 UMS $\leq^{\mathrm{p}}_{\mathrm{m}}$ GRAPH ISOMORPHISM.

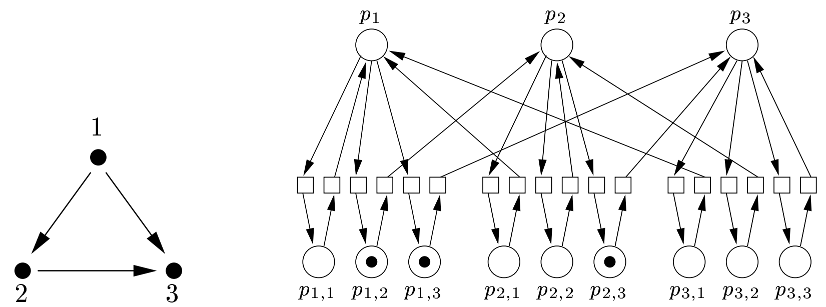
Proof. Let $N = \langle P, T, F, V, M_0 \rangle$. For a marking M of N, we interpret the marked net N as a labelled directed graph $G_M = \langle V_M, E_M, L_M \rangle$, where

1. $V_M = P \cup T$,
2. $\langle x, y \rangle \in E_M$ iff $\langle x, y \rangle \in F$,
3. $L_M(p) = M(p)$ for each $p \in P$ and $L_M(t) =$ "T" for all $t \in T$, and
4. $L_M(f) = V(f)$ for each $f \in F$.

It is obvious from the definition of G_M that two markings, M and M', are symmetric if and only if G_M and $G_{M'}$ are isomorphic. □

Lemma 4.6 GRAPH ISOMORPHISM $\leq^{\mathrm{p}}_{\mathrm{m}}$ IMS.

Proof. Suppose that we are given two (non-labelled) directed graphs, $G = \langle V, E \rangle$ and $G' = \langle V, E' \rangle$, with the same set of vertices (if they have a different number of vertices, they cannot be isomorphic and we can output a simple non-symmetric net and two different markings for it; if they have different



(a) A graph G. (b) The net and marking for G.

Figure 2: Reduction from a graph to net.

sets of vertices, any renaming of the vertices will do). We build the net $\hat{N} = \langle \hat{P}, \hat{T}, \hat{F}, \hat{V}, \hat{M}_0 \rangle$ as follows.

$$
\begin{aligned}
\hat{P} &= \{\hat{p}_v \mid v \in V\} \cup \{\hat{p}_{v,v'} \mid v, v' \in V\} \\
\hat{T} &= \left\{\hat{t}_{v,\langle v,v' \rangle} \mid v, v' \in V\right\} \cup \left\{\hat{t}_{\langle v,v' \rangle, v'} \mid v, v' \in V\right\} \\
\hat{F} &= \left\{\langle \hat{p}_v, \hat{t}_{v,\langle v,v' \rangle} \rangle \mid v, v' \in V\right\} \cup \left\{\langle \hat{t}_{v,\langle v,v' \rangle}, \hat{p}_{v,v'} \rangle \mid v, v' \in V\right\} \cup \\
&\quad \left\{\langle \hat{p}_{v,v'}, \hat{t}_{\langle v,v' \rangle, v'} \rangle \mid v, v' \in V\right\} \cup \left\{\langle \hat{t}_{\langle v,v' \rangle, v'}, \hat{p}_{v'} \rangle \mid v, v' \in V\right\} \\
\hat{V}(\hat{f}) &= 1 \text{ for all } \hat{f} \in \hat{F}
\end{aligned}
$$

The initial marking $\hat{M}_0$ is irrelevant, set it to be the empty marking.

For the graph G, we construct the corresponding marking $\hat{M}_G$ of $\hat{N}$ defined by

$$
\hat{M}_G(\hat{p}) = \begin{cases} 0 & \text{if } \hat{p} = \hat{p}_v \text{ for some } v \in V \\ 1 & \text{if } \hat{p} = \hat{p}_{v,v'} \text{ and } \langle v, v' \rangle \in E \\ 0 & \text{if } \hat{p} = \hat{p}_{v,v'} \text{ and } \langle v, v' \rangle \notin E \end{cases}
$$

and similarly $\hat{M}_{G'}$ for the graph G'. The idea of the construction is that the places of the form $\hat{p}_{v,v'}$ are used to represent the adjacency matrix of the graph under consideration. Figure 2(b) illustrates the construction by showing the net $\hat{N}$ and the corresponding marking for the graph in Fig. 2(a).

The automorphisms of $\hat{N}$ are exactly those that are generated by the group homomorphism $h : \mathrm{Sym}(V) \to \mathrm{Sym}(\hat{P} \cup \hat{T})$ such that $h(\pi)(\hat{p}_v) = \hat{p}_{\pi(v)}$, $h(\pi)(\hat{p}_{v,v'}) = \hat{p}_{\pi(v),\pi(v')}$, $h(\pi)(\hat{t}_{v,\langle v,v' \rangle}) = \hat{t}_{\pi(v),\langle \pi(v),\pi(v') \rangle}$ and $h(\pi)(\hat{t}_{\langle v,v' \rangle, v'}) = \hat{t}_{\langle \pi(v),\pi(v') \rangle, \pi(v')}$. That is, $\mathrm{Aut}(\hat{N}) = h(\mathrm{Sym}(V))$. As $\mathrm{Sym}(V)$ can be represented by two generators, the rotation $\pi_1 = \left(\begin{smallmatrix} v_1 & v_2 & v_3 & \cdots & v_{|V|-1} & v_{|V|} \\ v_2 & v_3 & v_4 & \cdots & v_{|V|} & v_1 \end{smallmatrix}\right)$ and the swapping of the first two elements $\pi_2 = \left(\begin{smallmatrix} v_1 & v_2 & v_3 & \cdots & v_{|V|} \\ v_2 & v_1 & v_3 & \cdots & v_{|V|} \end{smallmatrix}\right)$, the generators for $\mathrm{Aut}(\hat{N})$ are $h(\pi_1)$ and $h(\pi_2)$. Now it is easy to see that M_G and $M_{G'}$ are symmetric iff G and G' are isomorphic since $\mathrm{Aut}(\hat{N})$ corresponds to the group of all permutations on the vertex set V naturally extended to the adjacency matrix of a graph with the vertex set V. $\square$

We have thus obtained

$$\text{GRAPH ISOMORPHISM} \le_{\mathrm{m}}^{\mathrm{p}} \text{IMS} \le_{\mathrm{m}}^{\mathrm{p}} \text{UMS} \le_{\mathrm{m}}^{\mathrm{p}} \text{GRAPH ISOMORPHISM}$$

and as a consequence have the following.

Theorem 4.7 IMS *and* UMS *are both many-one equivalent to* GRAPH ISOMORPHISM.

Therefore, from the complexity theoretical point of view, pre-calculation of the automorphism group of a net does not provide any help for solving the problem of whether two markings are symmetric. However, in practice it is probably reasonable to compute the automorphism group of the net since it yields useful information. For instance, it may reveal that the net has no non-trivial automorphisms and thus the symmetry reduction method is of no use. Furthermore, knowing the automorphism group can assist in the choice of the integration algorithm since the performances of different algorithms depend on the order of the automorphism group (see [Schmidt 1999; 2000b]).

4.3 Canonical Representative Markings

An alternative for checking whether a symmetric marking has already been visited during the quotient reachability graph generation is to transform a newly generated marking into a representative marking.

Definition 4.8 *For a net N and for a marking M of N, a function* $\mathrm{repr} : \mathbb{M} \to \mathbb{M}$ *is a* representative function *if* $\mathrm{repr}(M) \equiv M$ *for all* $M \in \mathbb{M}$*.* repr *is* canonical *if* $\mathrm{repr}(M') = M$ *implies* $\mathrm{repr}(M'') = M$ *for all* $M'' \equiv M'$*.*

It is easy to see that having a canonical representative function would solve the marking symmetry problem because we could simply generate the canonical representative markings for the two markings in question and then check whether the representative markings are the same. Therefore, *calculating a canonical representative marking is at least as hard as answering to the graph isomorphism problem.* Fortunately the correctness of the symmetry reduction method does not depend crucially on the canonicity of repr. Therefore repr can be a heuristic algorithm that just tries to map the orbit $[M]$ into a set $\mathrm{repr}([M])$ as small as possible (see [Schmidt 1999; 2000b] for such an algorithm).

Assume however that we would like to have a canonical representative function repr. For this purpose we have to define which marking in an orbit is the canonical one. Perhaps the most obvious choice is to choose the lexicographically greatest (or smallest) marking in the orbit. In the following we study the complexity of finding such canonical markings.

For a net N, we implicitly assume an arbitrary total order $<_P$ on its places. We therefore have a lexicographical ordering for markings of N (also denoted by $<_P$) defined for all markings M, M' of N by

$$M <_P M' \Leftrightarrow (\exists p \in P)\big(M'(p) > M(p) \text{ and } (\forall p' <_P p)(M'(p') = M(p'))\big)$$

The following problem is now defined:

Problem 4.9 LEX-GREATEST MARKING. *Given a net N, its automorphism group* $\mathrm{Aut}(N)$ *and a marking M, find the lexicographically greatest marking symmetric to M.*

To classify the problem, we employ the problem CLIQUE SIZE asking the size of the largest clique in an undirected graph.

Lemma 4.10 CLIQUE SIZE $\leq_m^p$ LEX-GREATEST MARKING.

Proof. We use a construction resembling the one by Babai and Luks [1983, Section 3.1]. Given a non-labelled undirected graph $G = \langle V, E \rangle$, construct the net $\hat{N}$ and marking $\hat{M}_G$ for G as in the proof of Lemma 4.6. Now, assume an arbitrary total order $<_V$ on the set V of vertices. Define $UL(v) = \{\hat{p}_{v',v''} \mid v', v'' <_V v\}$ (the set of places corresponding to the edges between vertices that precede v, or, the upper left square down to v in the adjacency matrix of G). Define the total order on places of N to be such that the first $|V|^2$ places are the places of the form $\hat{p}_{v,v'}$, ordered in a way that the places in $UL(v)$ are before those in $UL(v')$ for all $v <_V v'$. Now the lex-greatest marking symmetric to $\hat{M}_G$ reveals the size of the largest clique in G. □

Since CLIQUE SIZE is known to be $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete [Krentel 1988; Papadimitriou 1995], we have the following.

Theorem 4.11 LEX-GREATEST MARKING *is* $\mathbf{FP}^{\mathbf{NP}[\log n]}$*-hard.*

In order to prove an upper bound for the LEX-GREATEST MARKING problem, we consider its decision version.

Problem 4.12 LEX-GREATER MARKING. *Given a net N, $\mathrm{Aut}(N)$ and two markings M and M', does there exist a marking M'' that (i) is lexicographically greater than or equals to M' and (ii) is symmetric to M?*

Lemma 4.13 LEX-GREATER MARKING *is* **NP***-complete.*

Proof. The problem is in **NP** because we can (i) guess a permutation σ of N, (ii) verify that σ is an automorphism of N, (iii) calculate $\sigma(M)$ and (iv) check whether $M' = \sigma(M)$ or $M' <_P \sigma(M)$, all in non-deterministic polynomial time.[2] LEX-GREATER MARKING is **NP**-hard because of the following. Take the construction in the proof of Lemma 4.10 to be the net. Suppose that we can say whether there is a marking that (i) is lexicographically greater than or equals to the marking in which the first k^2 places are marked and others are not and (ii) is symmetric to the marking $\hat{M}_G$ corresponding to a graph G. We can then tell whether the graph G has clique of size k or more, which is an **NP**-complete problem. □

Based on this we can prove the following.

Theorem 4.14 LEX-GREATEST MARKING *is in* $\mathbf{FP}^{\mathbf{NP}}$.

Proof. Let $m = \max_{p \in P}\{M(p)\}$ be the maximum number of tokens in the marking M. Then the representation of M is at least $\lceil \log_k m \rceil$ symbols long for some fixed k (the size of the Turing machine alphabet used) while the representation of the net N is at least of size $\mathcal{O}(|P|)$. We now can find and

[2]Note that we do not really need to consult the given group $\mathrm{Aut}(N)$ but can check whether the guessed permutation is an automorphism of N in deterministic polynomial time directly by using N.

fix the number of tokens of the first place in the lex-greatest symmetric marking by a binary search that calls at most $\lceil \log_k m \rceil$ times the LEX-GREATER MARKING oracle. After that, we can fix the number of the tokens in the second place similarly, and so on. Thus, we can find the lex-greatest symmetric marking with $\lceil \log_k m \rceil \cdot |P|$, a polynomial amount w.r.t. $\lceil \log_k m \rceil + \mathcal{O}(|P|)$, calls to an **NP** oracle. $\square$

It is currently open whether LEX-GREATEST MARKING is $\mathbf{FP}^{\mathbf{NP}[\log n]}$- or $\mathbf{FP}^{\mathbf{NP}}$-complete.

Remark 4.15 *The complexity* LEX-GREATEST MARKING *stays the same even if we do not know the automorphism group of the net.*

A note should be made that our choice for a canonical representative was probably not the most easily computable: according to Blass and Gurevich [1984], the lexicographically smallest element in an equivalence class can be in general harder to compute than an arbitrary canonical representative. However, as noted earlier, in our case computing any kind of canonical representative marking is at least as hard as answering to the graph isomorphism problem.

5 MARKING-STABILIZERS

For many markings it may be the case that some automorphisms map the marking to itself. We now demonstrate how such *marking-stabilizers* can be exploited and study what is the complexity of calculating them (cf. "self-symmetries" of Jensen [1995; 1996] and "state symmetry" in [Emerson and Sistla 1996; Gyuris and Sistla 1999]).

Definition 5.1 *The* stabilizer *of a marking M is*

$$\mathrm{Stab}(M) = \{\sigma \in \mathrm{Aut}(N) \mid \sigma(M) = M\} .$$

Clearly $\mathrm{Stab}(M)$ is a sub-group of $\mathrm{Aut}(N)$. The algorithm of Schmidt [2000a] can be used to compute marking-stabilizers.

One way to exploit marking-stabilizers is based on the following observation:

Lemma 5.2 *If $M\ [t\rangle\ M_1$, then $M\ [\sigma(t)\rangle\ \sigma(M_1)$ for all $\sigma \in \mathrm{Stab}(M)$.*

Proof. Directly by the fact that $M\ [t\rangle\ M_1 \Leftrightarrow \sigma(M)\ [\sigma(t)\rangle\ \sigma(M_1)$ holds for all $\sigma \in \mathrm{Aut}(N)$ and $\sigma(M) = M$ for a $\sigma \in \mathrm{Stab}(M) \subseteq \mathrm{Aut}(N)$. $\square$

Note that if we know the group $\mathrm{Stab}(M)$, then it is easy to check, given two transitions t and t', whether there is a $\sigma \in \mathrm{Stab}(M)$ such that $\sigma(t) = t'$. Assume that we are visiting a marking M during the quotient reachability graph generation. Now we have to check the enabledness of and fire only one transition per transition orbit under $\mathrm{Stab}(M)$ instead of all the transitions. If a transition in an orbit is enabled, then (and only then) all the transitions in it are, too. Furthermore, we know that all the transitions in the orbit will lead to mutually symmetric markings. We thus do not have to apply the marking symmetry test (or the canonization procedure) to each successor marking but to only one in the orbit.

Marking-stabilizers can also improve the "loop over all symmetries"-approach for the integration problem (recall Sec. 3). Consider a left coset $\sigma\,\mathrm{Stab}(M)$, where $\sigma \in \mathrm{Aut}(N)$. Now for each $\sigma' \in \sigma\,\mathrm{Stab}(M)$, $\sigma'(M) = \sigma(M)$. Thus it suffices to inspect only one symmetry per each left coset. Since $\mathrm{Stab}(M)$ is a sub-group of $\mathrm{Aut}(N)$, $\mathrm{Aut}(N)$ is divided into $\frac{|\mathrm{Aut}(N)|}{|\mathrm{Stab}(M)|}$ mutually disjoint left cosets. These facts were also noticed by Jensen [1995, page 92].

5.1 Complexity of Calculating Marking-Stabilizers

We formalize the following problem.

Problem 5.3 MARKING-STABILIZER. *Given a net N and a marking M of N, compute* $\mathrm{Stab}(M)$.

Theorem 5.4 MARKING-STABILIZER *and* GRAPH AUTOMORPHISMS *are many-one equivalent.*

Proof. We use the construction of Lemma 4.5 to reduce from MARKING-STABILIZER to GRAPH AUTOMORPHISMS. The automorphism group of G_M clearly corresponds to the stabilizer of the given marking M.

To reduce from GRAPH AUTOMORPHISMS to MARKING-STABILIZER, use the net $\hat{N}$ of Lemma 4.6. Now the stabilizer of the marking $\hat{M}_G$ for the given directed graph G is equivalent to $\mathrm{Aut}(G)$ when restricted to places of form $\hat{p}_v$. □

Remark 5.5 *The complexity of* MARKING-STABILIZER *remains the same even if we know the automorphism group of the net N.*

5.2 Canonical Representative Markings and Marking-Stabilizers

There is a connection between marking-stabilizers and canonical representative markings. Let repr be a canonical representative function for a net N.

Definition 5.6 *A left coset $\sigma\,\mathrm{Stab}(M)$, where $\sigma \in \mathrm{Sym}(P \cup T)$ such that $\sigma(M) = \mathrm{repr}(M)$, is called a* canonical labeling coset *of M.*

Canonical labeling cosets are desirable since they give both the canonical representative of a marking and also the stabilizer of the representative. Consequently, computing such cosets is a function problem at least as hard as GRAPH AUTOMORPHISMS.

A similar concept is used in the graph automorphism tool NAUTY tool by McKay [1990] which computes the automorphism group and the canonical form of a graph at the same time. See also [Babai and Luks 1983] for a string canonization algorithm.

6 SYMMETRIC COVERABILITY

We say that a marking M *covers* a marking M' if $M' \leq M$. In order to build a coverability graph [Finkel 1990] of a net, we extend markings to be functions of form $M : P \to (\mathbb{N} \cup \{\omega\})$, where ω is a symbol not in $\mathbb{N}$ and for all $x \in \mathbb{N} \cup \{\omega\}$, $x \leq \omega$. The coverability graph construction can be combined with the symmetry reduction method, see [Petrucci 1990]. We use the following definitions of Schmidt [2000a]:

Definition 6.1 *A marking M* symmetrically covers *a marking M', denoted by $M' \leqq M$, if there is a $\sigma \in \mathrm{Aut}(N)$ such that $M' \leq \sigma(M)$.*

Problem 6.2 SYMMETRIC COVERABILITY. *Given a net N and two of its markings, M and M', does M symmetrically cover M'?*

Schmidt [2000a] has extended his algorithm to solve the symmetric coverability problem.

Interestingly, the complexity of SYMMETRIC COVERABILITY jumps from graph isomorphism to **NP**-completeness, a phenomenon resembling that happening when we move from graph isomorphism to sub-graph isomorphism [Garey and Johnson 1979].

Theorem 6.3 SYMMETRIC COVERABILITY *is* **NP**-*complete.*

Proof. Obviously SYMMETRIC COVERABILITY is in **NP**. We show **NP**-hardness by reduction from the **NP**-complete problem CLIQUE asking if a graph $G = \langle V, E\rangle$ has a clique of size k or more. Construct the net $\hat{N}$ and the marking $\hat{M}_G$ for G as in the proof of Lemma 4.6. Let $\hat{M}'_G$ be a marking of $\hat{N}$ in which all the places of form $\hat{p}_{v,v'}$, where $v, v' \in V' \subseteq V$ such that $|V'| = k$, have one token and the other places are empty. Now clearly $\hat{M}_G$ symmetrically covers $\hat{M}'_G$ iff G has a clique of size k or more. $\square$

Remark 6.4 *Again, the complexity of* SYMMETRIC COVERABILITY *does not depend on whether we know the automorphism group of the net in question. Furthermore, it does not depend on the extension of markings with the ω symbol.*

6.1 Canonical Representative Markings and Symmetric Coverability

A way to solve the symmetric coverability problem would be to build a canonical representative function that solves the coverability problem at the same time:

Definition 6.5 *A canonical representative function* repr *is* suitable *for symmetric coverability if* $\mathrm{repr}(M') \leq \mathrm{repr}(M) \Leftrightarrow M' \leqq M$ *for all $M, M' \in \mathbb{M}$.*

Unfortunately, suitable representative functions do not always exist, as is shown in the next example and theorem.

Example 6.6 The function that chooses the lexicographically greatest marking in an orbit is *not* a suitable canonical representative function. For a counter-example, consider the net in Fig. 3 and assume the total order

$$p_i <_P p_j \Leftrightarrow i < j$$

between the places. Now the marking $M = 2p_0 + 2p_1 + 0p_2$ is its own representative $\text{repr}(M)$, while for $M' = 0p_0 + 1p_1 + 2p_2$ the representative is $\text{repr}(M') = 2p_0 + 0p_1 + 1p_2$. Now M symmetrically covers M' since $\sigma(M) = 0p_0 + 2p_1 + 2p_2 \geq M'$, where σ maps each p_i to $p_{i+1 \mod 3}$. But $\text{repr}(M') \leq \text{repr}(M)$ does not hold. ♣

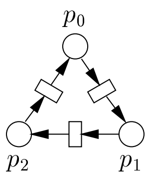


Figure 3: A net.

Theorem 6.7 *There exist nets for which suitable canonical representative functions do not exist.*

Proof. Assume that such functions exist for all nets. Consider again the net N in Fig. 3. Take the marking $M = 2p_0 + 2p_1 + 0p_2$ of N and any of its representatives, say $\text{repr}(M) = M$. Consider two other markings, $M_1 = 2p_0 + 1p_1 + 0p_2$ and $M_2 = 1p_0 + 2p_1 + 0p_2$. Clearly M symmetrically covers both M_1 and M_2. In order to repr to be suitable for symmetric coverability, it must be that $\text{repr}(M_1) = M_1$ and $\text{repr}(M_2) = M_2$ (other representatives lead to a situation in which place p_2 has one or more tokens and thus $\text{repr}(M)$ would not cover them). Now consider the marking $M' = 2p_0 + 1p_1 + 1p_2$ which symmetrically covers both markings M_1 and M_2. To repr to be suitable, it must be that $\text{repr}(M') = M'$ since other representatives do not cover $\text{repr}(M_1)$. But now $\text{repr}(M')$ does not cover $\text{repr}(M_2)$. Thus the initial assumption must be wrong and suitable canonical representative functions do not exist for all nets. □

7 CONCLUSIONS

In this paper we have addressed the computational complexity issues concerning the symmetry reduction method for Place/Transition-nets. Computing the automorphism group of a net was shown to be a task as hard as computing the automorphism group a graph. Although no polynomial time algorithm is known (or is expected to be found) for the task, it is not considered to be very hard in practice. The main problem in the symmetry reduction method, detecting whether two markings are symmetric, was proven to be equivalent to the GRAPH ISOMORPHISM problem under many-one reductions. Interestingly, this result does not depend on whether we know the automorphism group of the net in question or not. Building lexicographically greatest (smallest) canonical representative markings was shown to be a function problem lying somewhere between $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}^{\mathbf{NP}}$.

We have also discussed the use of marking-stabilizers of a marking (net's automorphisms that leave the marking intact) to improve the method. Computing the group of marking-stabilizers of a marking was classified to be equivalent to the GRAPH AUTOMORPHISMS problem.

As our last problem we have studied the symmetric coverability problem which combines the symmetry reduction method with the coverability graph approach. An interesting phenomenon occurred there: the symmetric coverability problem turned out to be an **NP**-complete problem instead of staying as hard as GRAPH ISOMORPHISM. Furthermore, we also found out that there exist nets for which the symmetric coverability problem and the canonical representative marking approach do not mix well.

Acknowledgments

The financial support of Helsinki Graduate School in Computer Science and Engineering (HeCSE) and the Academy of Finland (project no. 47754) is gratefully acknowledged.

References

BABAI, L. AND LUKS, E. M. 1983. Canonical labeling of graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*. ACM, 171–183.

BLASS, A. AND GUREVICH, Y. 1984. Equivalence relations, invariants, and normal forms. *SIAM Journal on Computing 13,* 4 (Nov.), 682–689.

BUTLER, G. 1991. *Fundamental Algorithms for Permutation groups*. Lecture Notes in Computer Science, vol. 559. Springer-Verlag, Berlin.

CLARKE, E. M., ENDERS, R., FILKORN, T., AND JHA, S. 1996. Exploiting symmetry in temporal logic model checking. *Formal Methods in System Design 9,* 1/2 (Aug.), 77–104.

EMERSON, E. A. AND SISTLA, A. P. 1996. Symmetry and model checking. *Formal Methods in System Design 9,* 1/2 (Aug.), 105–131.

FINKEL, A. 1990. The minimal coverability graph for Petri nets. In *11th Internation Conference on Application and Theory of Petri Nets*. 1–21.

FURST, M., HOPCROFT, J., AND LUKS, E. 1980. Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science*. IEEE, 36–41.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco.

GENRICH, H. J. 1991. Predicate/transition nets. In *High-level Petri Nets; Theory and Application*, K. Jensen and G. Rozenberg, Eds. Springer-Verlag, 3–43.

GYURIS, V. AND SISTLA, A. P. 1999. On-the-fly model checking under fairness that exploits symmetry. *Formal Methods in System Design 15,* 3 (Nov.), 217–238.

HUBER, P., JENSEN, A. M., JEPSEN, L. O., AND JENSEN, K. 1985. Towards reachability trees for high-level Petri nets. In *Advances in Petri Nets 1984*, G. Rozenberg, Ed. Lecture Notes in Computer Science, vol. 188. Springer-Verlag, 215–233.

HUBER, P., JENSEN, A. M., JEPSEN, L. O., AND JENSEN, K. 1991. Reachability trees for high-level Petri nets. In *High-level Petri Nets; Theory and Application*, K. Jensen and G. Rozenberg, Eds. Springer-Verlag, 319–350.

JENSEN, K. 1995. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use: Volume 2, Analysis Methods*. Monographs in Theoretical Computer Science. Springer-Verlag.

JENSEN, K. 1996. Condensed state spaces for symmetrical coloured Petri nets. *Formal Methods in System Design 9,* 1/2 (Aug.), 7–40.

JUNTTILA, T. 1999. Detecting and exploiting data type symmetries of algebraic system nets during reachability analysis. Research Report A57, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland. Dec.

KÖBLER, J., SCHÖNING, U., AND TORÁN, J. 1993. *The Graph Isomorphism Problem: Its Structural Complexity*. Progress in Theoretical Computer Science. Birkhäuser, Boston.

KREHER, D. L. AND STINSON, D. R. 1999. *Combinatorial Algorithms: Generation, Enumeration and Search*. CRC Press, Boca Raton, Florida.

KRENTEL, M. W. 1988. The complexity of optimization problems. *Journal of Computer and System Sciences 36,* 3 (June), 490–509.

MCKAY, B. D. 1990. Nauty user's guide (version 1.5). Tech. Rep. TR-CS-90-02, Computer Science Department, Australian National University.

MILLER, G. L. 1979. Graph isomorphism, general remarks. *Journal of Computer and System Sciences 18,* 2 (Apr.), 128–142.

PAPADIMITRIOU, C. H. 1995. *Computational Complexity*. Addison-Wesley, Reading, Massachusetts.

PETRUCCI, L. 1990. Combining Finkel's and Jensen's reduction techniques to build covering trees for coloured nets. *Petri Net Newsletter 36*, 32–36.

SCHMIDT, K. 1997. How to calculate symmetries of Petri nets. Tech. Rep. MATH-AL-8-1997, Technische Universität Dresden, Germany. Sept.

SCHMIDT, K. 1999. Integrating low level symmetries into reachability analysis. Informatik-Bericht 122, Humboldt-Universität zu Berlin, Institut für Informatik, Berlin, Germany.

SCHMIDT, K. 2000a. How to calculate symmetries of Petri nets. *Acta Informatica 36,* 7, 545–590.

SCHMIDT, K. 2000b. Integrating low level symmetries into reachability analysis. In *Tools and Algorithms for the Construction and Analysis of Systems; 6th International Conference, TACAS 2000,* S. Graf and M. Schwartzbach, Eds. Lecture Notes in Computer Science, vol. 1785. Springer, 315–330.

SCHMIDT, K. AND STARKE, P. H. 1991. An algorithm to compute the symmetries of Petri nets. *Petri Net Newsletter 40*, 25–30.

STARKE, P. H. 1991. Reachability analysis of Petri nets using symmetries. *Systems Analysis Modelling Simulation 8,* 4/5, 293–303.

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A46	Tuomas Aura Stateless connections. May 1997.
HUT-TCS-A47	Patrik Simons Towards Constraint Satisfaction through Logic Programs and the Stable Model Semantics. August 1997.
HUT-TCS-A48	Tuomas Aura On the structure of delegation networks. December 1997.
HUT-TCS-A49	Tomi Janhunen Non-Monotonic Systems: A Framework for Analyzing Semantics and Structural Properties of NMR. March 1998.
HUT-TCS-A50	Ilkka Niemelä (Ed.) Proceedings of the HeCSE Workshop on Emerging Technologies in Distributed Systems. March 1998.
HUT-TCS-A51	Kimmo Varpaaniemi On the Stubborn Set Method in Reduced State Space Generation. May 1998.
HUT-TCS-A52	Ilkka Niemelä, Torsten Schaub (Eds.) Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning. May 1998.
HUT-TCS-A53	Stefan Rönn Semantics of Semaphores. 1998.
HUT-TCS-A54	Antti Huima Analysis of Cryptographic Protocols via Symbolic State Space Enumeration. August 1999.
HUT-TCS-A55	Tommi Syrjänen A Rule-Based Formal Model For Software Configuration. December 1999.
HUT-TCS-A56	Keijo Heljanko Deadlock and Reachability Checking with Finite Complete Prefixes. December 1999.
HUT-TCS-A57	Tommi Junttila Detecting and Exploiting Data Type Symmetries of Algebraic System Nets during Reachability Analysis. December 1999.
HUT-TCS-A58	Patrik Simons Extending and Implementing the Stable Model Semantics. April 2000.
HUT-TCS-A59	Tommi Junttila Computational Complexity of the Place/Transition-Net Symmetry Reduction Method. April 2000.

ISBN 951-22-4984-7
ISSN 0783-5396