Journal of Universal Computer Science, vol. 7, no. 4 (2001), 307-326
submitted: 6/10/00, accepted: 4/4/01, appeared: 28/4/01 © Springer Pub. Co.

Computational Complexity of the Place/Transition-Net Symmetry Reduction Method

Tommi A. Junttila
(Laboratory for Theoretical Computer Science
Helsinki University of Technology, Finland
Tommi.Junttila@hut.fi)

Abstract: Computational complexity of the sub-tasks in the symmetry reduction method for Place/Transition-nets is studied. The 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. Deciding whether two markings are symmetric is shown to be a problem equivalent to the graph isomorphism problem. This remains to be the case even if a generator set for the automorphism group of the net is 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 be $\mathbf{FP}^{\mathbf{NP}[\log n]}$-hard but in $\mathbf{FP}^{\mathbf{NP}}$. 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 proven that canonical representative markings and the symmetric coverability problem cannot be combined in a straightforward way.

Key Words: Petri nets – symmetry – computational complexity
Category: F.2 [**Analysis of Algorithms and Problem Complexity**], D.2.4 [**Software Engineering**]: Program Verification

1 Introduction

Petri nets are a widely used formalism for modelling and analysis of distributed systems. Their success is based on the facts that they are relatively easy to understand, have a precise mathematical semantics and also a convenient graphical representation form. However, the most common analysis method, the reachability analysis (or state space exploration), suffers from the so-called state space explosion problem [Valmari 1998]. It essentially means that a net may have an exponential number of reachable markings (states) with respect to its size.

One way to alleviate the state space explosion problem is to exploit the symmetries (automorphisms) of the state space. These symmetries divide the state space into equivalence classes of mutually symmetric markings (called orbits). For many verification tasks, such as deadlock checking, it is sufficient to inspect only one marking in each (reachable) orbit. Thus a potentially much smaller quotient reachability graph consisting only of one (or few) markings per orbit can be constructed instead of the normal reachability graph. The symmetry reduction method was introduced in [Huber et al. 1984; 1991] for colored high-level Petri nets (i.e. nets in which tokens can have values associated with them). The method was applied to low-level nets (nets having only "black" tokens), the formalism used in this paper, in [Starke 1991] and further studied in [Schmidt and Starke 1991, Schmidt 1997; 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. (In high-level nets, the state space symmetries are usually produced by the symmetric use of the data values appearing in the tokens.) Schmidt and Starke have presented algorithms for solving many of the problems involved in the method [Schmidt and Starke 1991, Schmidt 1997; 2000a; 2000b]. However, unlike many other computational complexity aspects concerning Petri nets (see [Esparza 1998] for an introduction), the complexity issues of the sub-tasks appearing in the symmetry reduction method have not been addressed before.

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.

During the generation of the quotient reachability graph, the main task is to decide whether a marking symmetric to the newly generated marking has already been visited. To avoid the pair-wise symmetry comparison of already visited markings with the newly generated marking, 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 intuitively 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 $\mathbf{FP}^{\mathbf{NP}}$, inclusively.

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

As the last problem we consider the coverability problem under symmetries. A marking is said to cover another marking if each place in the marking has at least as many tokens as in the other marking. In Place/Transition-nets, the set of enabled transitions in a covering marking is a superset of the enabled transitions in the covered marking. This is exploited in the coverability graph generation [Finkel 1990], a technique that can be applied to check e.g. the boundedness of a net. Symmetries can be exploited also during the coverability graph generation and the symmetric coverability problem is: Given two markings of a net, is there 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 not, unfortunately, allow itself to be integrated into the canonical representative marking approach in a straightforward way.

1.1 Related Results

There are some related results concerning other system description formalisms.

First, [Clarke et al. 1996; 1998] consider the case where system states are given as vectors of state variables and the symmetry group acts on these by permuting the variable positions. The problem of deciding symmetricity of two states in this setting was first shown to be *at least as hard as* the graph isomorphism problem [Clarke et al. 1996], and then to be polynomially equivalent to the problems in the Luks equivalence class[1] [Clarke et al. 1998]. Furthermore, the problem of generating lexicographically smallest state symmetric to a given state is **NP**-hard [Babai and Luks 1983]. Note that the automorphism group of a Place/Transition-net acts in the same way on the markings (which can be seen as integer vectors). However, since it is a graph automorphism group (not an arbitrary group), we can prove in this paper that the marking symmetry problem is *as hard as* the graph isomorphism problem. In [Clarke et al. 1998] it is also shown that symmetries of a composition of parallel processes can be derived by constructing a corresponding hyper-graph and finding its automorphism group. In addition, it was shown in [Clarke et al. 1996] that there are symmetry groups for which the state equivalence (orbit) relation cannot be expressed by a polynomial-size Binary Decision Diagram (BDD).

Murφ is a verification system in which symmetries are described by the user by using special data types called scalar-sets. Permutations of the data values in scalar-sets then produce corresponding symmetries in the states. In [Ip and Dill 1996] it is shown that deciding whether two states of a Murφ program are symmetric is at least as hard as testing graph isomorphism.

Finally, for some related complexity theoretical results concerning a high-level Petri net formalism, see [Junttila 1999].

1.2 Outline

The paper is structured as follows. The necessary preliminaries are given in Section 2. Place/Transition-nets and their symmetries are defined in Section 3. 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 Section 4. The concept, use and computational complexity of marking-stabilizers is discussed in Section 5 while the symmetric coverability problem is studied in Section 6. Finally, we conclude in Section 7.

[1] Luks equivalence class is named in [Babai 1994] and contains many important group theoretical problems. The decision problems complete for it are still in **NP** but are believed to be harder than graph isomorphism.

2 Preliminaries

2.1 Computational Complexity Theory

See, e.g., [Garey and Johnson 1979, Papadimitriou 1994] for computational complexity theory in general. For simplicity, we will use the fairly standard polynomial time many-one (or Karp) reductions in this paper. For two decision problems (problems requiring a "no" or "yes" answer), A and B, we say that A *polynomial time many-one reduces* to B, denoted by $A \leq_{\mathrm{m}}^{\mathrm{p}} B$, if there is a polynomial time computable function R such that for all instances x of A, $R(x)$ is a "yes"-instance of B iff x is a "yes"-instance of A. If both $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ and $B \leq_{\mathrm{m}}^{\mathrm{p}} A$ 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 requiring an answer more elaborate than "no" or "yes", we use the following generalization of polynomial time many-one reductions. We say that a function problem f *polynomial time many-one reduces* to another function problem g, denoted by $f \leq_{\mathrm{m}}^{\mathrm{p}} g$, if there are polynomial time computable functions R and S such that for all instances x of f, (i) x has solutions in f iff $R(x)$ has solutions in g, and (ii) if y is a solution to $R(x)$ in g, then $S(y)$ is a solution to x in f. This reduction technique is similar to that in [Papadimitriou 1994] except that 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. The class **FP** is the class of function problems computable by 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. The n in $\mathbf{FP}^{\mathbf{NP}[\log n]}$ represents the fact that the number of queries to the oracle depends on the input size n. The classes $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}^{\mathbf{NP}}$ are important since many optimization problems such as Clique Size and Traveling Salesperson Problem, resp., are complete for them [Papadimitriou 1994].

2.2 Graph-Theoretical Problems

Since nets can be seen as directed labelled graphs, we use graph theoretical problems to classify the problems concerning net symmetries.

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 edge set is symmetric. Furthermore, it is *non-labelled* if the range of the labelling function is a unit set (all the 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 $v \in V_1$ 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 1. Graph Isomorphism. Given two labelled directed graphs, are they isomorphic?

It is easy to see, based on results in [Miller 1979], that the 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 paper. Graph Isomorphism 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). In fact, if Graph Isomorphism is an **NP**-complete problem, then the polynomial-time hierarchy will collapse to its second level [Boppana et al. 1987, Goldreich et al. 1986]. This is generally considered to be a very unlikely event to happen. 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 $\mathrm{Aut}(G)$ and is a group under the function composition operation $\circ$.

Problem 2. Graph Automorphisms. Given a graph G, find a set of generators for $\mathrm{Aut}(G)$ (see the text below for discussion on permutation groups and generators sets).

Graph Automorphisms is a function problem that is polynomially equivalent to Graph Isomorphism in the sense that if either has a polynomial time algorithm, then (and only then) both have [Luks 1993]. Again, the complexity of Graph Automorphisms is the same for graphs and labelled directed graphs.

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 $\circ$. A sub-group of $\mathrm{Sym}(A)$ for a set A is called a permutation group. A set of generators for a permutation group is a set of permutations (called generators) such that any permutation in the group can be expressed as a composition of generators. *In this paper it is assumed that permutation groups are given by means of their generator sets.* For instance, we will now on write "find $\mathrm{Aut}(G)$" instead of "find a set of generators for $\mathrm{Aut}(G)$" and "given $\mathrm{Aut}(G)$" instead of "given a set of generators for $\mathrm{Aut}(G)$". It is well-known 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. Using this normal form, we can test in polynomial time whether a permutation belongs to the group [Furst et al. 1980] and whether two elements $a, b \in A$ belong to the same orbit (i.e. whether there is a permutation in the group that maps a to b). Furthermore, each sub-group of $\mathrm{Sym}(A)$ has a generator set consisting of at most $|A| - 1$ generators [Jerrum 1986]. For permutation group algorithms, see, e.g. [Butler 1991, Kreher and Stinson 1999].

3 Place/Transition-Nets and their Symmetries

We are now ready to present Place/Transition-nets and their symmetries. The presentation in this section is based on [Starke 1991, Schmidt and Starke 1991, Schmidt 1997; 2000a].

3.1 Place/Transition-Nets

A *Place/Transition-net* (or a P/T-net) is a tuple $N = \langle P, T, F, W, M_0 \rangle$, 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. $W : F \rightarrow \mathbb{N}_+$ maps each arc in F with a *multiplicity* (we define that $W(\langle x, y \rangle) = 0$ if $\langle x, y \rangle \notin F$) and
5. $M_0 : P \rightarrow \mathbb{N}$ is the *initial marking* of N.

A *marking* of N is a function $M : P \rightarrow \mathbb{N}$ 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 instance, if we have the places p_1, p_2 and p_3, the marking $M = \{p_1 \mapsto 1, p_2 \mapsto 3, p_3 \mapsto 0\}$ can be denoted by the formal sum $1p_1 + 3p_2 + 0p_3$. Dropping the places with multiplicity 0 and omitting unit multiplicities, M can also be written as $p_1 + 3p_2$. For two markings, M and M', $M \leq M'$ iff $M(p) \leq M'(p)$ for all $p \in P$. A transition $t \in T$ is *enabled* in a marking M if $W(\langle p, t \rangle) \leq M(p)$ for all $p \in P$. If t is enabled in M, it may *fire* and transform M into M' defined by $M'(p) = M(p) - W(\langle p, t \rangle) + W(\langle t, p \rangle)$ 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. A *symmetry* (an automorphism) of N is a permutation $\sigma \in \mathrm{Sym}(P \cup T)$ which

1. respects node type: $\sigma(P) = P$ and $\sigma(T) = T$;
2. respects the flow relation, i.e., $\langle x, y\rangle \in F \Leftrightarrow \langle \sigma(x), \sigma(y)\rangle \in F$ for all $x, y \in P \cup T$; and
3. respects the arc multiplicities: $W(\langle x, y\rangle) = W(\langle \sigma(x), \sigma(y)\rangle)$ 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 acts on markings of N by $\sigma(M) = M \circ \sigma^{-1}$, or equivalently, $(\sigma(M))(\sigma(p)) = M(p)$ for all $p \in P$. We say that two markings, M and M', of N are *symmetric*, denoted by $M \equiv M'$, if $\sigma(M) = M'$ for some $\sigma \in \mathrm{Aut}(N)$. The set of markings symmetric to a marking M is the equivalence class $[M] = \{\sigma(M) \mid \sigma \in \mathrm{Aut}(N)\}$ (the *orbit* of M under $\mathrm{Aut}(N)$). It is these equivalence classes and the following lemma that are exploited in the symmetry reduction method.

Lemma 4 [Starke 1991]. *Let σ be a symmetry, t a transition and M, M' two markings of N. Then $M\ [t\rangle\ M' \Leftrightarrow \sigma(M)\ [\sigma(t)\rangle\ \sigma(M')$.*

That is, symmetric states have a symmetric behavior. 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 instance, it is relatively easy to see (by applying Lemma 4) that a marking appears in a quotient reachability graph if a symmetric marking appears in the reachability graph, and vice versa. Furthermore, if a marking is a deadlock marking (no transitions are enabled in it), then and only then all the markings symmetric to it are also deadlock markings. Thus a quotient reachability graph has a deadlock iff the reachability graph has. For more on these properties and advanced algorithms for temporal logic model checking under symmetries, see [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 2000b]:

Problem 5. 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 problem [Schmidt 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 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 1. Consider the variant of Genrich's railroad system net [Genrich 1991] shown in Fig. 1(a). All the arc multiplicities in the net equal to 1 and are not drawn here or in any subsequent figures. Its reachability graph is shown in Fig. 1(b). The group $\mathrm{Aut}(N)$ is generated by the rotation

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

and the swapping of train identities

$$\sigma_{\mathrm{swap}} = \begin{pmatrix} U_{a0} & \cdots & U_{a5} & U_{b0} & \cdots & U_{b5} & V_0 & \cdots & V_5 & t_{a0} & \cdots & t_{a5} & t_{b0} & \cdots & t_{b5} \\ U_{b0} & \cdots & U_{b5} & U_{a0} & \cdots & U_{a5} & V_0 & \cdots & V_5 & t_{b0} & \cdots & t_{b5} & t_{a0} & \cdots & t_{a5} \end{pmatrix}.$$

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 $\sigma_{\mathrm{swap}}(\sigma_{\mathrm{rot}}(M_0)) = \sigma_{\mathrm{swap}}(U_{a1} + U_{b4} + V_2 + V_5) = M$. 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.

4 Complexity of the Fundamental Sub-Problems

4.1 Computing Net Automorphisms

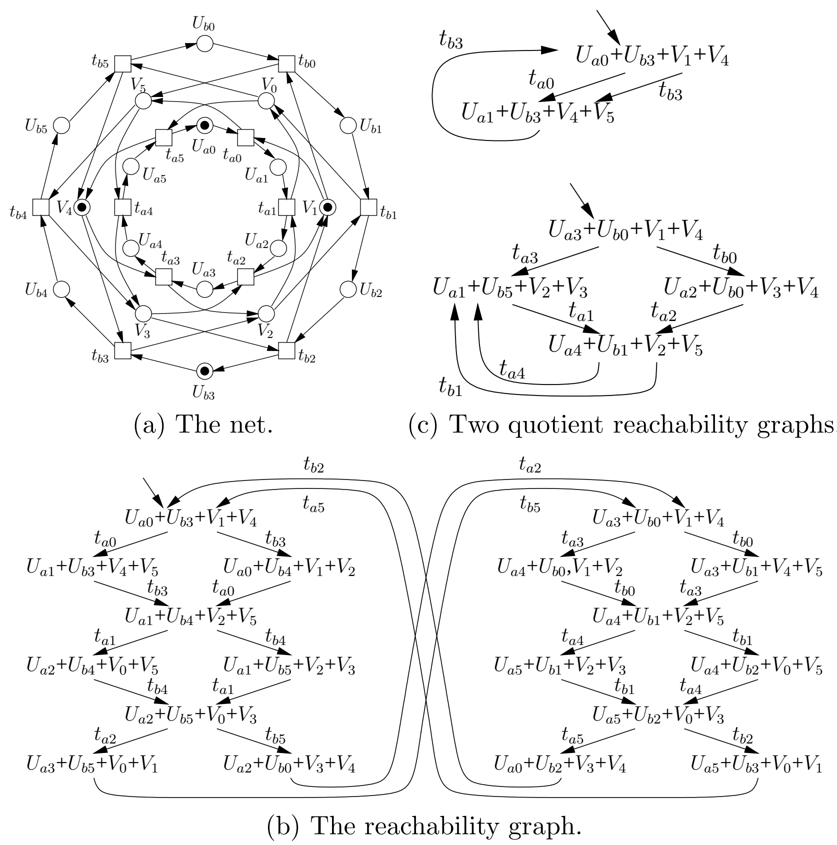
The first problem is to find the automorphism group of a net.

Problem 6. NET AUTOMORPHISMS. Given a net N, compute $\mathrm{Aut}(N)$.

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

Theorem 7. NET AUTOMORPHISMS *is many-one equivalent to* GRAPH AUTOMORPHISMS.

Proof. We first reduce from GRAPH AUTOMORPHISM to NET AUTOMORPHISMS. Given a directed graph $G = \langle V, E\rangle$, construct the net $N = \langle P, T, F, W, M_0\rangle$ where $P = V$, $T = E$, $F = \{\langle v, \langle v, v'\rangle\rangle \mid \langle v, v'\rangle \in E\} \cup \{\langle\langle v, v'\rangle, v'\rangle \mid \langle v, v'\rangle \in E\}$ and $W(f) = 1$ 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



(a) The net. (c) Two quotient reachability graphs.

(b) The reachability graph.

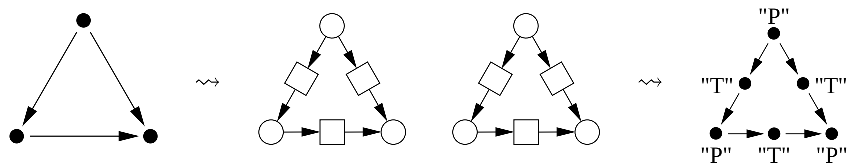
Figure 1: A net for a railroad system and its (quotient) reachability graph(s).

is 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 of the graph is the automorphism group of the net. See Fig. 2 for simple examples (arc multiplicities and edge labels are omitted for simplicity). □

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 8. Uninformed Marking Symmetry (UMS). Given a net N and two markings of N, are the markings symmetric?



(a) From graph to net. (b) From net to graph.

Figure 2: Reductions between graphs and nets.

Problem 9. 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 10. UMS $\leq^{\mathrm{p}}_{\mathrm{m}}$ Graph Isomorphism.

Proof. Let $N = \langle P, T, F, W, M_0 \rangle$. 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. $L_M(f) = W(f)$ for each $f \in F$.

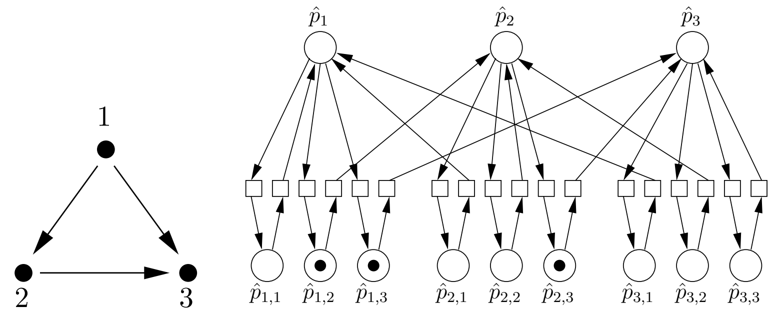
It is obvious from the definition of G_M that two markings, M and M', are symmetric if and only if G_M and $G_{M'}$ are isomorphic. □

Lemma 11. 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 sets of vertices, any renaming of the vertices will do). We build the net $\hat{N} = \langle \hat{P}, \hat{T}, \hat{F}, \hat{W}, \hat{M}_0 \rangle$ as follows.

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

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



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

Figure 3: Reduction from a graph to a net.

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 3(b) illustrates the construction by showing the net $\hat{N}$ (transition names are omitted) and the corresponding marking for the graph in Fig. 3(a).

The automorphisms of $\hat{N}$ are exactly those that are generated by the homomorphism $h : \mathrm{Sym}(V) \to \mathrm{Sym}(\hat{P} \cup \hat{T})$ such that (i) $(h(\pi))(\hat{p}_v) = \hat{p}_{\pi(v)}$, (ii) $(h(\pi))(\hat{p}_{v,v'}) = \hat{p}_{\pi(v),\pi(v')}$, (iii) $(h(\pi))(\hat{t}_{v,\langle v,v'\rangle}) = \hat{t}_{\pi(v),\langle \pi(v),\pi(v')\rangle}$ and (iv) $(h(\pi))(\hat{t}_{\langle v,v'\rangle,v'}) = \hat{t}_{\langle \pi(v),\pi(v')\rangle,\pi(v')}$. 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 $\hat{M}_G$ and $\hat{M}_{G'}$ 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. That is, if the vertices of G can be permuted in a way that the adjacency matrix of G becomes equal to the adjacency matrix of G', then (and only then) can the marking $\hat{M}_G$ be permuted by $\mathrm{Aut}(\hat{N})$ to become equal to $\hat{M}_{G'}$. For instance, consider the marking $\hat{p}_{1,2} + \hat{p}_{1,3} + \hat{p}_{2,3}$ shown in Fig. 3(b) corresponding to the adjacency matrix of the graph in Fig. 3(a). Applying the generator $h(\pi_1)$ to the marking, we obtain the marking $\hat{p}_{2,3} + \hat{p}_{2,1} + \hat{p}_{3,1}$. This marking corresponds to the adjacency matrix of the graph obtained from that in Fig. 3(a) by replacing the vertex "1" with "2", "2" with "3" and "3" with "1". Obviously this graph is isomorphic to the one in Fig. 3(a). □

We have thus established

$$\text{Graph Isomorphism} \leq^{\mathrm{p}}_{\mathrm{m}} \text{IMS} \leq^{\mathrm{p}}_{\mathrm{m}} \text{UMS} \leq^{\mathrm{p}}_{\mathrm{m}} \text{Graph Isomorphism}$$

and as a consequence have the following.

Theorem 12. 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 2000b].

4.3 Canonical Representative Markings

An alternative approach 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 13. For a net, a function of form $repr : \mathbb{M} \to \mathbb{M}$ is a *representative function* if $repr(M) \equiv M$ for all markings $M \in \mathbb{M}$. The function *repr* is *canonical* if $repr(M') = M$ implies $repr(M'') = M$ 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* [Clarke et al. 1996, Ip and Dill 1996, Schmidt 2000b]. Therefore *repr* can be a heuristic algorithm that just tries to map the orbit $[M]$ into a set $repr([M])$ as small as possible (see [Schmidt 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 the lexicographical total order for markings of N (also denoted by $<_P$) such that $M <_P M'$ iff

$$(\exists p \in P)\big(M'(p) > M(p) \wedge (\forall p' <_P p)(M'(p') = M(p'))\big).$$

The following problem is now defined:

Problem 14. 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 15. Clique Size $\leq^{\mathrm{p}}_{\mathrm{m}}$ Lex-Greatest Marking.

Proof. We use a construction resembling one in [Babai and Luks 1983, Section 3.1]. Given a non-labelled undirected graph $G = \langle V, E\rangle$ (the edge set is assumed to be reflexive, i.e. all vertices have a self-loop), construct the net $\hat{N}$ and marking $\hat{M}_G$ for G as in the proof of Lemma 11. 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 $\hat{N}$ 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 because the first k^2 places are marked in the marking iff G has a clique of size k or more. □

As Clique Size is known to be an $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete problem [Krentel 1988, Papadimitriou 1994], we have the following.

Theorem 16. 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 17. 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 18. 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. Given a graph G, construct the net $\hat{N}$ as in the proof of Lemma 15. 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 each have one token and the rest are empty 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. □

[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.

Based on this we can prove the following.

Theorem 19. 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 $\Theta(|P|)$. We now can find and 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. $\lceil \log_k m \rceil + \Theta(|P|)$, calls to an **NP** oracle. □

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

Remark. The lower and upper bounds for LEX-GREATEST MARKING, given in Theorems 16 and 19, stay the same even if the automorphism group of the net is not given as input.

A note should be made that our choice for a canonical representative was probably not the most easily computable: [Blass and Gurevich 1984] shows that 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. "state symmetry" in [Emerson and Sistla 1996, Gyuris and Sistla 1999] and "self-symmetries" in [Jensen 1995; 1996]).

Definition 20. 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 in [Schmidt 2000a] can be used to compute marking-stabilizers.

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

Lemma 21. *If* $M\ [t\rangle\ M_1$*, then* $M\ [\sigma(t)\rangle\ \sigma(M_1)$ *for each* $\sigma \in \mathrm{Stab}(M)$.

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 $\sigma \in \mathrm{Stab}(M) \subseteq \mathrm{Aut}(N)$. □

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 an automorphism $\sigma \in \mathrm{Stab}(M)$ such that $\sigma(t) = t'$. The group $\mathrm{Stab}(M)$ also partitions the set T of transitions into disjoint orbits, the orbit of a transition t being $[t]_{\mathrm{Stab}(M)} = \{\sigma(t) \mid \sigma \in \mathrm{Stab}(M)\}$. These orbits are also easy to compute, given the group $\mathrm{Stab}(M)$. 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 Section 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. As $\mathrm{Stab}(M)$ is a subgroup 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 in [Jensen 1995, page 92].

5.1 Complexity of Calculating Marking-Stabilizers

We formalize the following problem.

Problem 22. MARKING-STABILIZER. Given a net N and a marking M of N, compute $\mathrm{Stab}(M)$.

Theorem 23. MARKING-STABILIZER *and* GRAPH AUTOMORPHISMS *are many-one equivalent.*

Proof. To reduce from MARKING-STABILIZER to GRAPH AUTOMORPHISMS, we use the construction of Lemma 10. 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 11. 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. 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 24. A left coset $\sigma\,\mathrm{Stab}(M)$, where $\sigma \in \mathrm{Sym}(P \cup T)$ such that $\sigma(M) = \mathit{repr}(M)$, is called a *canonical labelling coset* of M.

Canonical labelling 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 [McKay 1990] which computes the automorphism group and a 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 [Karp and Miller 1969, 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 in [Schmidt 2000a]:

Definition 25. 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 26. SYMMETRIC COVERABILITY. Given a net N and two of its markings, M and M', does M symmetrically cover M'?

Schmidt has extended his algorithm for testing the symmetricity of two markings to solve the SYMMETRIC COVERABILITY problem [Schmidt 2000a].

Interestingly, the complexity of SYMMETRIC COVERABILITY jumps from GRAPH ISOMORPHISM to **NP**-completeness, a phenomenon resembling that happening when moving from GRAPH ISOMORPHISM to SUB-GRAPH ISOMORPHISM [Garey and Johnson 1979].

Theorem 27. 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 an undirected graph $G = \langle V, E \rangle$ has a clique of size k or more. Again, the graph G is assumed to have a reflexive edge set meaning that all vertices have a self-loop. Construct the net $\hat{N}$ and the marking $\hat{M}_G$ for G as in the proof of Lemma 11. 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. □

Remark. 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.

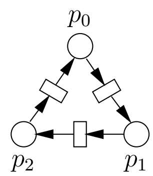


Figure 4: A net with no suitable canonical representative function.

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 28. A canonical representative function *repr* is *suitable* for symmetric coverability if $repr(M') \leq repr(M) \Leftrightarrow M' \leqq M$ 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 2. The function that chooses the lexicographically greatest marking in an orbit is *not* a suitable canonical representative function for all nets. For a counter-example, consider the net in Fig. 4 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 $repr(M)$, while for $M' = 0p_0 + 1p_1 + 2p_2$ the representative is $repr(M') = 2p_0 + 0p_1 + 1p_2$. 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 $repr(M') \leq repr(M)$ does not hold.

Theorem 29. *There exists 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. 4. Take the marking $M = 2p_0 + 2p_1 + 0p_2$ and any of its representatives, say $repr(M) = M$. 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 $repr(M_1) = M_1$ and $repr(M_2) = M_2$ (other representatives lead to a situation in which place p_2 has one or more tokens and thus $repr(M)$ 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 $repr(M') = M'$ since other representatives do not cover $repr(M_1)$. But now $repr(M')$ does not cover $repr(M_2)$. 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 of 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.

Acknowledgements

The author wishes to thank Ilkka Niemelä, Keijo Heljanko and the anonymous referees for their comments on this work. The financial support of Helsinki Graduate School in Computer Science and Engineering (HeCSE), the Academy of Finland (project no. 47754) and Tekniikan Edistämissäätiö ("Foundation of Technology") is also gratefully acknowledged.

References

[Babai 1994] Babai, L.: "Automorphism groups, isomorphism, reconstruction"; Technical Report TR-94-10, University of Chicago, Department of Computer Science (1994). Also as chapter 27 of the "Handbook of Combinatorics", North-Holland (1995).

[Babai and Luks 1983] Babai, L., Luks, E. M.: "Canonical labeling of graphs"; Proc. Fifteenth Annual ACM Symposium on Theory of Computing, ACM (1983), 171–183.

[Blass and Gurevich 1984] Blass, A., Gurevich, Y.: "Equivalence relations, invariants, and normal forms"; SIAM Journal on Computing, 13, 4 (1984), 682–689.

[Boppana et al. 1987] Boppana, R. B., Hastad, J., Zachos, S.: "Does co-NP have short interactive proofs?" Information Processing Letters, 25, 2 (1987), 127–132.

[Butler 1991] Butler, G.: "Fundamental Algorithms for Permutation groups"; volume 559 of Lecture Notes in Computer Science, Springer-Verlag, Berlin (1991).

[Clarke et al. 1998] Clarke, E. M., Emerson, E. A., Jha, S., Sistla, A. P.: "Symmetry reductions in model checking"; in "Computer Aided Verification (CAV'98)", eds. Hu, A. J. and Vardi, M. Y., volume 1427 of Lecture Notes in Computer Science, Springer, Berlin (1998), 147–158.

[Clarke et al. 1996] Clarke, E, M., Enders, R., Filkorn, T., Jha, S.: "Exploiting symmetry in temporal logic model checking"; Formal Methods in System Design, 9, 1/2 (1996), 77–104.

[Emerson and Sistla 1996] Emerson, E.A., Sistla, A.P.: "Symmetry and model checking"; Formal Methods in System Design, 9, 1/2 (1996), 105–131.

[Esparza 1998] Esparza, J.: "Decidability and complexity of Petri net problems — an introduction"; in "Lectures on Petri Nets I: Basic Models", Reisig, W., Rozenberg, G., eds., volume 1491 of Lecture Notes in Computer Science, Springer (1998), 374–428.

[Finkel 1990] Finkel, A.: "The minimal coverability graph for Petri nets"; Proc. 11th Internation Conference on Application and Theory of Petri Nets (1990), 1–21.

[Furst et al. 1980] Furst, M., Hopcroft, J., Luks, E.: "Polynomial-time algorithms for permutation groups"; Proc. 21st Annual Symposium on Foundations of Computer Science, IEEE (1980), 36–41.

[Garey and Johnson 1979] Garey, M. R., Johnson, D. S.: "Computers and Intractability: A Guide to the Theory of NP-Completeness"; W. H. Freeman and Company, San Francisco (1979).

[Genrich 1991] Genrich, H. J.: "Predicate/transition nets"; in "High-level Petri Nets; Theory and Application", Jensen, K., Rozenberg, G., eds., Springer-Verlag (1991), 3–43.

[Goldreich et al. 1986] Goldreich, O., Micali, S., Wigderson, A.: "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design"; Proc. 27th Annual Symposium on Foundations of Computer Science, IEEE (1986), 174–187.

[Gyuris and Sistla 1999] Gyuris, V., Sistla, A. P.: "On-the-fly model checking under fairness that exploits symmetry"; Formal Methods in System Design, 15, 3 (1999), 217–238.

[Huber et al. 1984] Huber, P., Jensen, A. M., Jepsen, L. O., Jensen, K.: "Towards reachability trees for high-level Petri nets"; in "Advances in Petri Nets 1984", Rozenberg, G., ed., volume 188 of Lecture Notes in Computer Science Springer-Verlag (1985), 215–233.

[Huber et al. 1991] Huber, P., Jensen, A. M., Jepsen, L. O., Jensen, K.: "Reachability trees for high-level Petri nets"; in "High-level Petri Nets; Theory and Application", Jensen, K., Rozenberg, G., eds., Springer-Verlag (1991), 319–350.

[Ip and Dill 1996] Ip, C. N. and Dill, D. L.: "Better verification through symmetry"; Formal Methods in System Design, 9, 1/2 (1996), 41–76.

[Jensen 1995] Jensen, K.: "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use: Volume 2, Analysis Methods"; Monographs in Theoretical Computer Science, Springer-Verlag (1995).

[Jensen 1996] Jensen, K.: "Condensed state spaces for symmetrical coloured Petri nets"; Formal Methods in System Design, 9, 1/2 (1996), 7–40.

[Jerrum 1986] Jerrum, M.: "A Compact Representation for Permutation Groups"; Journal of Algorithms, 7, 1 (1986), 60–78.

[Junttila 1999] Junttila, T.: "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 (1999).

[Karp and Miller 1969] Karp, R. M., Miller, R. E.: "Parallel Program Schemata"; Journal of Computer and System Sciences, 3, 2 (1969). 147–195.

[Köbler et al. 1993] Köbler, J., Schöning, U., Torán, J.: "The Graph Isomorphism Problem: Its Structural Complexity"; Progress in Theoretical Computer Science, Birkhäuser, Boston (1993).

[Kreher and Stinson 1999] Kreher, D. L., Stinson, D. R.: "Combinatorial Algorithms: Generation, Enumeration and Search"; CRC Press, Boca Raton, Florida (1999).

[Krentel 1988] Krentel, M. W.: "The complexity of optimization problems"; Journal of Computer and System Sciences, 36, 3 (1988). 490–509.

[Luks 1993] Luks, E. M.: "Permutation groups and polynomial-time computation"; Technical Report CIS 93–01, Department of Computer and Information Science, University of Oregon (1993). Also published in "Groups and Computation", L. Finkelstein and W. M. Kantor, Eds. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, DIMACS (1993).

[McKay 1990] McKay, B. D.: "Nauty user's guide (version 1.5)"; Technical Report TR-CS-90-02, Computer Science Department, Australian National University (1990).

[Miller 1979] Miller, G. L.: "Graph isomorphism, general remarks"; Journal of Computer and System Sciences, 18, 2 (1979), 128–142.

[Papadimitriou 1994] Papadimitriou, C. H.: "Computational Complexity". Addison-Wesley, Reading, Massachusetts (1994).

[Petrucci 1990] Petrucci, L.: "Combining Finkel's and Jensen's reduction techniques to build covering trees for coloured nets"; Petri Net Newsletter, 36 (1990), 32–36.

[Schmidt 1997] Schmidt, K.: "How to calculate symmetries of Petri nets"; Technical Report MATH-AL-8-1997, Technische Universität Dresden, Germany (1997).

[Schmidt 2000a] Schmidt, K.: "How to calculate symmetries of Petri nets"; Acta Informatica, 36, 7 (2000), 545–590.

[Schmidt 2000b] Schmidt, K.: "Integrating low level symmetries into reachability analysis"; in "Tools and Algorithms for the Construction and Analysis of Systems; 6th International Conference, TACAS 2000", Graf, S., Schwartzbach, M., eds., volume 1785 of Lecture Notes in Computer Science, Springer (2000), 315–330.

[Schmidt and Starke 1991] Schmidt, K., Starke, P. H.: "An algorithm to compute the symmetries of Petri nets"; Petri Net Newsletter, 40 (1991), 25–30.

[Starke 1991] Starke, P. H.: "Reachability analysis of Petri nets using symmetries"; Systems Analysis Modelling Simulation, 8, 4/5 (1991), 293–303.

[Valmari 1998] Valmari, A.: "The state explosion problem"; in "Lectures on Petri Nets I: Basic Models", Reisig, W., Rozenberg, G., eds., volume 1491 of Lecture Notes in Computer Science, Springer (1998), 429–528.