Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 75
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 75
Espoo 2002 HUT-TCS-A75

NEW CANONICAL REPRESENTATIVE MARKING ALGORITHMS FOR PLACE/TRANSITION-NETS

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 75
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 75
Espoo 2002 HUT-TCS-A75

NEW CANONICAL REPRESENTATIVE MARKING ALGORITHMS FOR PLACE/TRANSITION-NETS

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 951-22-6218-5
ISSN 1457-7615

Otamedia Oy
Espoo 2002

Abstract: Symmetries of a Place/Transition-net can be exploited during the reachability analysis by considering only one representative marking in each orbit induced by the symmetries. In this report, three new algorithms for transforming a marking into a symmetric canonical representative marking are described. All the algorithms depend on the precalculation of a Schreier-Sims representation for the symmetry group of the net in question. The first algorithm uses a black box graph canonizer algorithm to produce a canonical version of the characteristic graph associated with a marking and then derives the canonical representative marking from it. The second algorithm is a backtrack search in the Schreier-Sims representation, pruning the search with the marking in question and its stabilizers found during the search. The third algorithm combines the first and second one by pruning the search in the Schreier-Sims representation with an ordered partition obtained with a standard preprocessing technique applied in graph isomorphism algorithms.

Keywords: Reachability analysis, Place/Transition-nets, symmetry

CONTENTS

1 INTRODUCTION

Symmetries of a Place/Transition-net produce symmetries into its state space [Starke 1991]. These state space symmetries can be exploited during the reachability analysis by considering only one (or few) marking(s) in each set of symmetric markings. This may result in substantial savings in both memory and time requirements of the reachability analysis. The main task during the generation of the reduced reachability graph is to decide whether a marking symmetric to the newly generated one is already visited during the search. This can be accomplished by either (i) comparing the new marking pairwisely with each already visited one, or (ii) transforming each generated marking into a canonical representative marking and storing only these into the reduced reachability graph. Some algorithms for these tasks are presented in [Schmidt 2000a; 2000b], while the computational complexity of the tasks is analyzed in [Junttila 2001].

This report describes three new algorithms for producing canonical representative markings. All the algorithms presented require that the automorphism group of the net is known. This is in contrast to some algorithms described in [Schmidt 2000a; 2000b]. The fact that all the algorithms presented in this report depend on precalculation of a Schreier-Sims representation for the symmetry group of the net is not a serious drawback. This is because it is beneficial to first compute the symmetry group of the net in order to see if there are any non-trivial symmetries, i.e., to see whether the symmetry reduction method can help at all. In addition, the performance of symmetry reduction algorithms may depend on the size of the symmetry group, see [Schmidt 2000b] and Sect. 6.2, and thus knowing it may help in selecting an appropriate algorithm.

The first algorithm presented in Sect. 3 uses a black box graph canonizer algorithm to produce a canonical representative for a marking. First, a characteristic graph is assigned to the marking. Characteristic graphs have the property that the characteristic graphs of two markings are isomorphic if and only if the markings are symmetric. Furthermore, the isomorphisms between the characteristic graphs correspond exactly to the symmetries transforming the markings to each other. The canonical version of the characteristic graph of a marking is then obtained by applying a black box graph canonizer, and finally the canonical representative for the marking is obtained by using an isomorphism between the characteristic graph and its canonical version. In [Junttila 2002], a similar algorithm was described for high-level Petri nets and related formalisms.

The second algorithm, presented in Sect. 4, is a backtrack search in the Schreier-Sims representation of the symmetry group. The algorithm returns the smallest marking produced by symmetries that are "compatible" with the marking in question. The search is pruned (i) by considering only symmetries that are "compatible" with the marking, (ii) by using the smallest already found symmetric marking, and (iii) by exploiting the stabilizers of the marking (which are found during the search). This algorithm is a variant of the standard backtrack search algorithms developed in the computational group theory, see e.g. [Butler 1991]. However, the compatibility definition between symmetries and markings is, to author's knowledge, novel.

The third algorithm presented in Sect. 5 combines the techniques used in Sects. 3 and 4 by "opening" the black box graph canonizer. A standard preprocessing technique of existing graph isomorphism algorithms (see e.g. [McKay 1981; Kreher and Stinson 1999]) is used to produce an ordered partition of the marking in question in a symmetry respecting way. The partition is then used to prune the backtrack search in the Schreier-Sims representation by considering only symmetries that are compatible with the partition.

1.1 Related Work

Some symmetry reduction algorithms for Place/Transition-nets are described in [Schmidt 2000a; 2000b]. The first algorithm, "iterating the symmetries", applies all the symmetries to the new marking and checks whether the resulting marking has already been visited during the reduced reachability graph construction. The facts that (i) the symmetries are stored in a Schreier-Sims representation and (ii) the set of already visited markings is stored as a prefix sharing decision tree, are exploited to prune the set of symmetries considered. The second algorithm, "iterating the states", pairwisely checks the newly generated marking with each already visited marking for symmetry by using the algorithm described in [Schmidt 2000a]. The set of necessary symmetry checks is reduced by using symmetry respecting hash functions. The third algorithm, "canonical representatives", computes a (non-canonical) representative for the newly generated marking. This is done by a limited search with greedy heuristics in the Schreier-Sims representation of symmetries, trying to find the lexicographically smallest symmetric marking. The second algorithm in this report can be seen as a complete, canonical version of this algorithm augmented with some pruning techniques.

In addition to Place/Transition-nets, symmetries are also exploited in the state space analysis of other formalisms. For the symmetry reduction method in general, see the papers in the volume 9, numbers 1–2, of the Formal Methods in System Design journal. For temporal logic model checking under symmetry, see [Emerson and Sistla 1996; 1997; Gyuris and Sistla 1999].

In [Clarke et al. 1996; Clarke et al. 1998], symmetries are defined to be permutations acting on bit vectors by permuting the element positions. This is very similar to the way the symmetries of Place/Transition-nets act on the markings. The difference to the approach discussed in this report is that the state sets are manipulated symbolically by using Binary Decision Diagrams (BDDs). Thus the symmetry related calculations, such as computing a representative for a state, are also performed by using BDDs. If explicit state enumeration were used instead of BDDs, the algorithms presented in this report could probably be applied.

In the Murφ system [Ip and Dill 1996] and in high-level Petri nets [Chiola et al. 1991; Jensen 1996; Junttila 1999], the symmetries are produced by permuting the values of data types. Usually, the symmetries in these formalisms don't have to be represented explicitly but are defined by declaring some data types to be permutable (e.g., the scalar set data types in Murφ). Despite these differences to Place/Transition-nets, the algorithms proposed in this report have some similarities to the existing ones for these formalisms.

1. An algorithm similar to the first algorithm in this report is described in

[Junttila 2002]. In both algorithms, a characteristic graph for a marking is first constructed. A canonical form for the graph is then obtained by using a black box graph canonizer, and the canonical representative marking is then obtained from it. The algorithms differ in the way the characteristic graphs are defined and in the way the canonical representative marking is obtained from the canonical version of the characteristic graph.
2. The obvious state canonization algorithm enumerating through all the states that are symmetric to the given state is not usually very effective for the above mentioned formalisms, see e.g. [Ip 1996]. The second algorithm in this report uses pruning techniques that make this approach work reasonably well for Place/Transition-nets.
3. Partitions are used to prune the set of symmetries that have to be considered during the checking whether two states are symmetric or during the canonization of states not only in the third algorithm in this report but also in many other algorithms, e.g. [Huber et al. 1985; Ip 1996; Sistla et al. 2000; Junttila 2002]. They mainly differ in the way the partitions are produced. The algorithm in this report is to author's knowledge the first one that can work under general symmetries (i.e., symmetries that are not produced by direct products of symmetric and/or cyclic groups like those induced by scalar sets). This is achieved through the use of a Schreier-Sims representation for the symmetry group and the novel compatibility definition between permutations and partitions.

Some other canonical representative algorithms are also presented in [Chiola et al. 1991; Lorentsen and Kristensen 2001].

2 PLACE/TRANSITION-NETS AND THEIR SYMMETRIES

First, some standard definitions of Place/Transition-nets are given. These are based on [Starke 1991; Schmidt 2000a; 2000b]. Formally, a *Place/Transition-net* (or simply a P/T-net) is a tuple

$$N = \langle P, T, F, W, M_0 \rangle,$$

where

1. P is a finite, nonempty set of *places*,
2. T is a finite, nonempty set of *transitions* such that $P \cap T = \emptyset$,
3. $F \subseteq (P \times T) \cup (T \times P)$ is the *flow-relation* (the set of *arcs*),
4. $W : F \to \mathbb{N} \setminus \{0\}$ associates each arc in F with a positive *multiplicity* (*weight*), and
5. $M_0 : P \to \mathbb{N}$ is the *initial marking* of N. A *marking* of N is a multiset on P, i.e., a function $M : P \to \mathbb{N}$, and the *set of all markings* of N is denoted by $\mathbb{M}$.

The multiplicity function W is implicitly extended to $(P \times T) \cup (T \times P) \to \mathbb{N}$ by $W(\langle x, y \rangle) = 0$ if $\langle x, y \rangle \notin F$. A marking M can also be denoted by the formal sum $\sum_{p \in P} M(p)'p$. For instance, if the set of places is $P =$

$\{p_1, p_2, p_3\}$, the marking $M = \{p_1 \mapsto 1, p_2 \mapsto 3, p_3 \mapsto 0\}$ can be denoted by the formal sum $1'p_1 + 3'p_2 + 0'p_3$. Dropping the places with multiplicity 0 and omitting unit multiplicities, M can also be written as $p_1 + 3'p_2$. For two markings, M and M', define that $M \leq M'$ if and only if $M(p) \leq M'(p)$ for each $p \in P$. A transition $t \in T$ is *enabled* in a marking M if $W(\langle p, t\rangle) \leq M(p)$ for each $p \in P$. 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 each $p \in P$. This is denoted by $M\ [t\rangle\ M'$.

A *symmetry* (an *automorphism*) of the net N is a permutation σ of $P \cup T$ that

- respects the node type: $\sigma(P) = P$ and $\sigma(T) = T$,
- respects the flow relation: $\langle x, y\rangle \in F \Leftrightarrow \langle \sigma(x), \sigma(y)\rangle \in F$, and
- respects the arc multiplicities: $W(\langle x, y\rangle) = W(\langle \sigma(x), \sigma(y)\rangle)$ for each $\langle x, y\rangle \in F$.

The set of *all symmetries of* N (the automorphism group of N) is denoted by $\mathrm{Aut}(N)$ and forms a group under the function composition operator $\circ$. A symmetry $\sigma \in \mathrm{Aut}(N)$ *acts* on the markings of N so that a marking M is mapped to the marking $\sigma(M)$ defined by $(\sigma(M))(p) = M(\sigma^{-1}(p))$ for each place $p \in P$. Equivalent definitions for the action are: $(\sigma(M))(\sigma(p)) = M(p)$ for each place p and $\sigma(M) = M \circ \sigma^{-1}$. Two markings, M_1 and M_2, are *symmetric under a subgroup* G *of* $\mathrm{Aut}(N)$ if there is a symmetry $\sigma \in G$ such that $\sigma(M_1) = M_2$. This is denoted by $M_1 \equiv_G M_2$. By the group properties of G, $\equiv_G$ is an equivalence relation on the set $\mathbb{M}$ of all markings and the equivalence class of a marking M is called the *G-orbit of M* and is denoted by $[M]_G$. In the case G is understood from the context, one may simply speak of symmetric markings and orbits, and omit the subscript G.

Symmetries of the net produce symmetries to the state space of the net [Starke 1991]: for each symmetry σ it holds that

$$M_1\ [t\rangle\ M_2 \Leftrightarrow \sigma(M_1)\ [\sigma(t)\rangle\ \sigma(M_2)$$

meaning that symmetric markings have symmetric successor markings. Thus for many verification tasks, such as finding deadlocks, the successor markings can be "redirected" to symmetric ones during the reachability graph generation, resulting in a *quotient reachability graph* that can be exponentially smaller than the original reachability graph, see e.g. [Clarke et al. 1996; Ip and Dill 1996; Jensen 1996]. With some extensions, temporal logic model checking by using quotient reachability graphs is also possible, see e.g. [Emerson and Sistla 1996; 1997; Gyuris and Sistla 1999]. Formally, a quotient reachability graph (QRG) is a labeled transition system $\langle Q, \Delta, M_0'\rangle$, where (i) $M_0' \equiv M_0$, and (ii) $Q \subseteq \mathbb{M}$ and $\Delta \subseteq Q \times T \times Q$ fulfill the following:

1. M_0' is in Q,
2. if $M \in Q$ and $M\ [t\rangle\ M'$, then (i) $\langle M, t, M''\rangle \in \Delta$ for one M'' such that $M'' \equiv M'$ and (ii) $M'' \in Q$, and
3. nothing else is in Q or in Δ.

Note that QRGs are not necessarily unique, i.e., there may be many QRGs for a given net and symmetry group G. Alg. 2.1 gives a QRG generation

Algorithm 2.1 A quotient reachability graph generation algorithm

1: Choose any M'_0 such that $M_0 \equiv M'_0$
2: Set $Q = \{M'_0\}$
3: Set $W = \{M'_0\}$
4: Set $\Delta = \emptyset$
5: **while** $W \neq \emptyset$ **do**
6:     Take any $M \in W$ and set $W = W \setminus \{M\}$
7:     **for all** $M\ [t\rangle\ M'$ **do**
8:         Choose any M'' such that $M' \equiv M''$
9:         Set $\Delta = \Delta \cup \{\langle M, t, M'' \rangle\}$
10:         **if** $M'' \notin Q$ **then**
11:             Set $Q = Q \cup \{M''\}$
12:             Set $W = W \cup \{M''\}$
13: Return $\langle Q, T, \Delta, M'_0 \rangle$

algorithm. The crucial line in the algorithm is the line 8, where a marking symmetric to the successor marking is selected. The goal is to obtain as small QRGs as possible, i.e., QRGs containing only one marking from each reachable orbit. There are two ways to achieve this.

1. The marking M' is pairwisely compared for symmetry with all markings in the set Q of already visited markings. If a marking symmetric to M' is found in Q, then M'' is defined to be that marking, otherwise M'' is defined to be M'. Symmetry respecting hash functions can be used to pruned the set of markings in Q that have to be tested for symmetry [Schmidt 2000a].
2. The marking M' is transformed into a *canonical representative marking* and M'' is defined to be that marking. Formally, a function *repr* : $\mathbb{M} \to \mathbb{M}$ is a *representative marking function* if *repr*$(M) \equiv M$ holds for each marking $M \in \mathbb{M}$. A representative marking function *repr* is *canonical* if $M_1 \equiv M_2$ implies *repr*$(M_1) =$ *repr*(M_2). In this case, the marking *repr*(M) is called the *canonical representative marking* of M (under *repr*). In this approach the initial marking has be canonized as well, i.e., M'_0 is defined to be *repr*(M_0) in line 1.

This report contributes to the second approach. Since the problem of deciding whether a marking is symmetric to another is as hard as the graph isomorphism problem [Junttila 2001], both of the approaches contain tasks for which no polynomial time algorithms are currently known. Luckily, the approaches can be approximated by (i) using a sound but incomplete marking symmetry check in the first one, and (ii) by using a non-canonical representative marking function in the second one. Using such an approximation may result in that more than one marking in a reachable orbit is visited during the search and thus the quotient reachability graph may not be of minimal size. Hence the space consumption (and sometimes the time consumption, too) may grow compared to an exact approach.

Another concept used in the rest of the report is that of a stabilizer. Given a subgroup G of $\mathrm{Aut}(N)$, the *stabilizer subgroup of a marking M in G* is $\mathrm{Stab}(G, M) = \{\sigma \in G \mid \sigma(M) = M\}$, i.e., the set of all symmetries of N in G that fix the marking M.

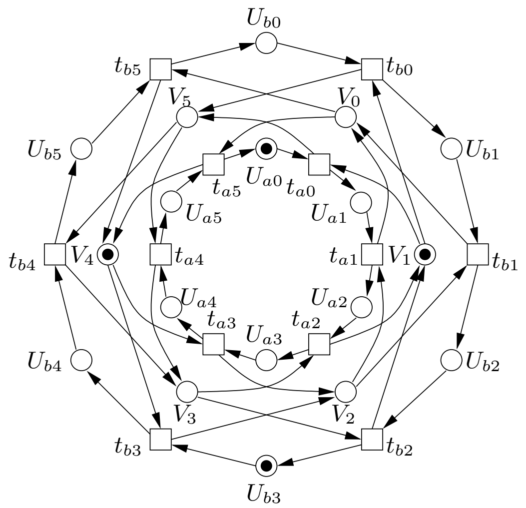


Figure 1: A variant of Genrich's railroad net.

Example 2.1 Consider the variant of Genrich's railroad system net [Genrich 1991] shown in Fig. 1. The places of the net are drawn as circles, transitions as rectangles, and the arcs between them as directed edges. All the arc multiplicities in the net equal to 1 and are not drawn here or in any subsequent figures. The black filled circles (*tokens*) in the figure describe the initial marking $U_{a0} + U_{b3} + V_1 + V_4$ of the net. The automorphism group $\mathrm{Aut}(N)$ of the net is generated by the rotation

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

of the railroad sections and the swapping of train identities a and b:

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

The group $\mathrm{Aut}(N)$ has 12 elements. Now the initial marking $M_0 = U_{a0} + U_{b3} + V_1 + V_4$ is symmetric (under $\mathrm{Aut}(N)$) to the marking $M = U_{a4} + U_{b1} + V_2 + V_5$ as $(\sigma_{\mathrm{swap}} \circ \sigma_{\mathrm{rot}})(M_0) = \sigma_{\mathrm{swap}}(\sigma_{\mathrm{rot}}(M_0)) = \sigma_{\mathrm{swap}}(U_{a1} + U_{b4} + V_2 + V_5) = M$. The orbit $[M_0]_{\mathrm{Aut}(N)}$ of M_0 consists of the markings

$$\begin{array}{ll} M_0, & U_{a1} + U_{b4} + V_2 + V_5, \\ U_{a2} + U_{b5} + V_0 + V_3, & U_{a3} + U_{b0} + V_1 + V_4, \\ U_{a4} + U_{b1} + V_2 + V_5, \text{ and } & U_{a5} + U_{b2} + V_0 + V_3. \end{array}$$

The reachability graph of the net is shown in Fig. 2, while Fig. 3 shows two quotient reachability graphs for the net, the left one being minimal in the sense that it contains only one marking from each orbit. The stabilizer group of the initial marking M_0 has two elements,

$$\mathrm{Stab}(\mathrm{Aut}(N), M_0) = \{\mathbf{I}, \sigma_{\mathrm{swap}} \circ \sigma_{\mathrm{rot}} \circ \sigma_{\mathrm{rot}} \circ \sigma_{\mathrm{rot}}\},$$

where $\mathbf{I}$ denotes the identity permutation. ♣

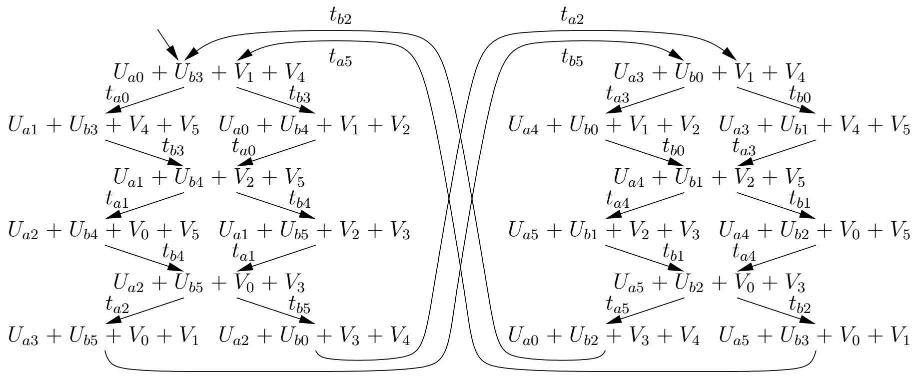


Figure 2: The reachability graph of the net in Fig. 1.

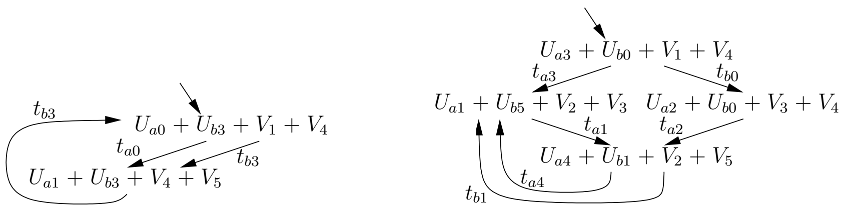


Figure 3: Two quotient reachability graphs for the net in Figure 1.

2.1 The Schreier-Sims Representation

Although a permutation group on a set of n elements may have up to $n!$ permutations, there are representations for permutation groups that have size polynomial in n. The following text describes one such standard representation that has some useful properties exploited later in this report. For more on permutation group algorithms, see [Butler 1991]. The presentation here is based on [Kreher and Stinson 1999].

Assume a finite set X and a permutation group G on X. For instance, X may be the set $P \cup T$ and G the group $\mathrm{Aut}(N)$ for a P/T-net $N = \langle P, T, F, W, M_0 \rangle$. Assume that $|X| = n$ and order the elements in X in any order $\beta = [x_1, x_2, \ldots, x_n]$. Let

$$\begin{aligned} G_0 &= G \\ G_1 &= \{g \in G_0 \mid g(x_1) = x_1\} \\ G_2 &= \{g \in G_1 \mid g(x_2) = x_2\} \\ &\vdots \\ G_n &= \{g \in G_{n-1} \mid g(x_n) = x_n\}. \end{aligned}$$

The groups $G_0, G_1, \ldots, G_n$ are subgroups of G such that

$$G = G_0 \geq G_1 \geq \cdots \geq G_n = \{\mathbf{I}\}$$

where $\mathbf{I}$ denotes the identity permutation. Note that a permutation $g \in G_i$, $0 \leq i \leq n$, fixes each element $x_1, \ldots, x_i$. For each $1 \leq i \leq n$, let $[x_i]_{G_{i-1}} = \{g(x_i) \mid g \in G_{i-1}\}$ denote the orbit of x_i under G_{i-1}. Assume that $[x_i]_{G_{i-1}} = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\}$ for some $1 \leq n_i \leq n$. For each $1 \leq j \leq n_i$, choose a $h_{i,j} \in G_{i-1}$ such that $h_{i,j}(x_i) = x_{i,j}$ and let $U_i = \{h_{i,1}, h_{i,2}, \ldots, h_{i,n_i}\}$.

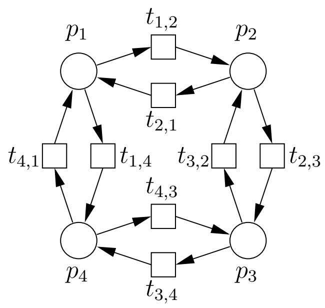


Figure 4: A toy example net.

Now U_i is a left transversal of G_i in G_{i-1}, i.e., $h_{i,j} \circ G_i \neq h_{i,k} \circ G_i$ for $j \neq k$ and $G_{i-1} = h_{i,1} \circ G_i \cup \cdots \cup h_{i,n_i} \circ G_i$, where $h \circ G_i$ denotes the left coset $\{h \circ g \mid g \in G_i\}$. The structure $\vec{G} = [U_1, U_2, \ldots, U_n]$ is a *Schreier-Sims representation* of the group G. Each element in $g \in G$, and only those, can be uniquely written as a composition $g = h_1 \circ h_2 \circ \cdots \circ h_n$, where $h_i \in U_i$, and thus the order of G equals to $|U_1|\,|U_2| \cdots |U_n|$. The ordering $\beta = [x_1, x_2, \ldots, x_n]$ is called the *base* of the representation. It can be and is assumed from now on that each U_i contains the identity permutation $\mathbf{I}$. As each U_i contains at most $n-i+1$ permutations, there are at most $n(n+1)/2$ permutations in the Schreier-Sims representation $\vec{G} = [U_1, U_2, \ldots, U_n]$.[1] Many operations, such as testing whether a permutation belongs to the group, can be performed in polynomial time by using Schreier-Sims representations. Furthermore, given a generating set of permutations for a group, the Schreier-Sims representation for the group can be calculated in polynomial time.

The ground sets in [Schmidt 2000a; 2000b] are actually Schreier-Sims representations. Thus the algorithm for computing the symmetries of a net presented in [Schmidt 2000a] produces a Schreier-Sims representation of the symmetry group.

Example 2.2 Consider the net in Fig. 4. Its automorphism group, call it G, under the base

$$\beta = [p_1, p_2, p_3, p_4, t_{1,2}, t_{2,1}, t_{2,3}, t_{3,2}, t_{3,4}, t_{4,3}, t_{4,1}, t_{1,4}]$$

has a Schreier-Sims representation $\vec{G} = [U_1, U_2, \ldots, U_{|P|+|T|}]$, where

$$\begin{aligned}
U_1 &= \left\{ \begin{array}{l} h_{1,1} = \mathbf{I} \\ h_{1,2} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_2 & p_3 & p_4 & p_1 & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} & t_{1,2} & t_{2,1} \end{pmatrix} \\ h_{1,3} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_3 & p_4 & p_1 & p_2 & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} \end{pmatrix} \\ h_{1,4} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_4 & p_1 & p_2 & p_3 & t_{4,1} & t_{1,4} & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} \end{pmatrix} \end{array} \right\}, \\
U_2 &= \left\{ \begin{array}{l} h_{2,1} = \mathbf{I} \\ h_{2,2} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_1 & p_4 & p_3 & p_2 & t_{1,4} & t_{4,1} & t_{4,3} & t_{4,2} & t_{3,2} & t_{2,3} & t_{2,1} & t_{1,2} \end{pmatrix} \end{array} \right\}, \text{ and} \\
U_i &= \{\mathbf{I}\} \text{ for } 3 \leq i \leq |P| + |T|.
\end{aligned}$$

[1] A more compact representation consisting of at most $n-1$ permutations could also be used instead [Jerrum 1986].

Therefore, $|G| = 8$. ♣

2.2 Place Valuations and Compatible Permutations

In addition to the standard Schreier-Sims representation definitions above, some new concepts are needed in the rest of the article.

To facilitate the understanding of the following concepts, a Schreier-Sims representation $\vec{G} = [U_1, \ldots, U_n]$ of a permutation group G under a base $\beta = [x_1, \ldots, x_n]$ can be seen as a tree. The levels of the tree correspond to the base of the representation and each node at a level i has $|U_i|$ children at the level $i + 1$, the edges to the children being labeled with the permutations in U_i. For instance, Fig. 5 shows (a prefix of) the tree corresponding to the Schreier-Sims representation in Ex. 2.2. Consider a path in the

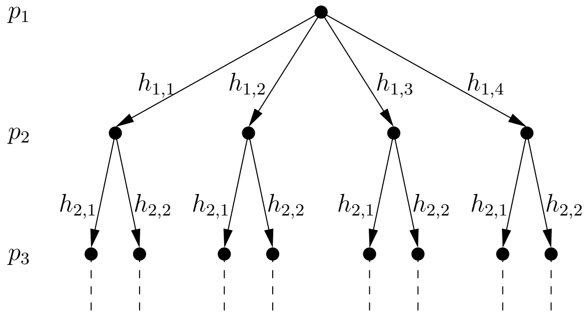


Figure 5: Schreier-Sims representation seen as a tree.

tree starting from the root and ending in a node v at a level i. Composing the labels of the edges in the path defines the corresponding permutation in $g \in U_1 \circ \cdots \circ U_{i-1}$. Thus the full paths ending in leaf nodes of the tree define exactly the permutations in the group. The node v has $|U_i|$ child nodes, and extending the path to any of them defines an extension permutation of g which is in $g \circ U_i$. The set $\{g(h(x_i)) \mid h \in U_i\}$ is now the set of $|U_i|$ possible images of the ith base element x_i under all the permutations corresponding to the paths going through the node v. If the elements in this set can be distinguished in some way, some of the child nodes can be pruned away during a search in the tree. The concepts needed for the pruning process considered in this report are discussed below and the pruning itself is formulated in Def. 2.3 of compatibility.

First, a *place valuation* is a function *pval* : $P \to \mathbb{N}$ assigning each place a natural number. Observe that the definition is exactly the same as for markings, a different name is only used in order to avoid confusions later. The action of permutations in $\mathrm{Aut}(N)$ on place valuations is defined similarly to that on markings.

Second, a *multiset selector* is a function from nonempty multisets over natural numbers to nonempty sets of natural numbers such that each number in the image set has a non-zero multiplicity in the argument multiset. That is, if *select* is a multiset selector and $n \in \textit{select}(m)$, then $m(n) \geq 1$. For instance, the *trivial* multiset selector is $\textit{select}_{\text{trivial}} = \{n \mid m(n) \geq 1\}$, e.g. $\textit{select}_{\text{trivial}}(3'2 + 2'4 + 2'5 + 4'7) = \{2, 4, 5, 7\}$. For a better example, define the *minimal element* multiset selector $\textit{select}_{\min}$ such that

$select_{\min}(m) = \{n\}$, where n is the smallest number that has non-zero multiplicity in m. Now $select_{\min}(3'2 + 2'4 + 2'5 + 4'7) = \{2\}$. Similarly, the *maximal element* multiset selector $select_{\max}$ would give $select_{\max}(3'2 + 2'4 + 2'5 + 4'7) = \{7\}$. Also define the *minimal element with minimal frequency* multiset selector $select_{\text{minminfreq}}$ such that $select_{\text{minminfreq}}(m) = \{n\}$, where n is the smallest number among those that have the smallest non-zero multiplicity in m. For example, $select_{\text{minminfreq}}(3'2 + 2'4 + 2'5 + 4'7) = \{4\}$. Similarly, the *maximal element with minimal frequency* multiset selector $select_{\text{maxminfreq}}$ would give $select_{\text{maxminfreq}}(3'2 + 2'4 + 2'5 + 4'7) = \{5\}$. A *function* multiset selector is a multiset selector for which the image set always contains exactly one element. All the other multiset selectors above except $select_{\text{trivial}}$ clearly fulfill this condition.

In the following definition, the above discussed pruning procedure is formulated by defining which permutations corresponding to the full paths in the tree survive the pruning. Such permutations will be called compatible. Formally, assume a fixed multiset selector *select*, a subgroup G of $\mathrm{Aut}(N)$, and a Schreier-Sims representation $\vec{G} = [U_1, \ldots, U_{|P|+|T|}]$ of G under a base $\beta = [p_{\beta,1}, \ldots, p_{\beta,|P|}, t_{\beta,1}, \ldots, t_{\beta,|T|}]$ in which the places are enumerated before the transitions.

Definition 2.3 *A permutation* $g_1 \circ \cdots \circ g_{|P|} \circ g_{|P|+1} \cdots \circ g_{|P|+|T|} \in G$, *where* $g_i \in U_i$, *is* compatible *with a place valuation pval if*

$$pval((g_1 \circ \cdots \circ g_{i-1} \circ g_i)(p_{\beta,i})) \in select\left(\sum_{h \in U_i} 1'pval((g_1 \circ \cdots \circ g_{i-1} \circ h)(p_{\beta,i}))\right)$$

holds for each $1 \leq i \leq |P|$ *(when* $i = 1$, $g_1 \circ \cdots \circ g_{i-1} = \mathbf{I}$*).*

The intuitive explanation for the above definition in the tree model is the following. First, in a node at level i, the images of the ith base element in the child nodes are computed. From the values assigned to these images by the place valuation, a subset is then selected by the multiset selector. All the child nodes in which the value assigned to the image of the ith base element by the place valuation is not in the subset, are then pruned away. As each node has at least one child that is not pruned away, there always is at least one permutation compatible with the place valuation.

Example 2.4 Recall the net in Fig. 4 and the Schreier-Sims representation of its automorphism group G described in Ex. 2.2. Assume a place valuation $pval = \{p_1 \mapsto 1, p_2 \mapsto 0, p_3 \mapsto 0, p_4 \mapsto 0\}$ and the minimal element multiset selector $select_{\min}$. Now

$$\begin{aligned}
select_{\min}\left(\sum_{h \in U_1} 1'pval(h(p_{\beta,1}))\right) &= \\
select_{\min}\left(1'pval(p_1) + 1'pval(p_2) + 1'pval(p_3) + 1'pval(p_4)\right) &= \\
select_{\min}\left(1'1 + 1'0 + 1'0 + 1'0\right) &= \\
select_{\min}\left(3'0 + 1'1\right) &= \{0\}
\end{aligned}$$

and thus $pval(\hat{g}_1(p_1)) = 0$ must hold for any permutation $\hat{g} = \hat{g}_1 \circ \cdots \circ \hat{g}_{12}$, $\hat{g}_i \in U_i$ for each $1 \leq i \leq 12$, that is compatible with *pval*. This requirement is fulfilled by $h_{1,2}$, $h_{1,3}$ and $h_{1,4}$.

If $\hat{g}_1 = h_{1,2}$, then

$$\begin{aligned} select_{\min}\left(\sum_{h \in U_2} 1'pval(h_{1,2}(h(p_{\beta,2})))\right) &= \\ select_{\min}\left(1'pval(p_3) + 1'pval(p_1)\right) &= \\ select_{\min}\left(1'1 + 1'0\right) &= \{0\} \end{aligned}$$

and thus $pval(h_{1,2}(\hat{g}_2(p_2))) = 0$ must hold for any permutation $\hat{g} = h_{1,2} \circ \hat{g}_2 \cdots \circ \hat{g}_{12}$ that is compatible with *pval*. This requirement is fulfilled by $h_{2,1}$. Because $U_i = \{\mathbf{I}\}$ for $i \geq 3$, the symmetry $h_{1,2} \circ h_{2,1} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_2 & p_3 & p_4 & p_1 & t_{2,3} & \cdots \end{pmatrix}$ is compatible with *pval*.

Similar computations show that the other permutations compatible with *pval* are

$$\begin{aligned} h_{1,3} \circ h_{2,1} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_3 & p_4 & p_1 & p_2 & t_{3,4} & \cdots \end{pmatrix}, \\ h_{1,3} \circ h_{2,2} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_3 & p_2 & p_1 & p_4 & t_{3,2} & \cdots \end{pmatrix}, \text{ and} \\ h_{1,4} \circ h_{2,2} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_4 & p_3 & p_2 & p_1 & t_{4,3} & \cdots \end{pmatrix}. \end{aligned}$$

To sum up, there are 4 permutations that are compatible with *pval*.

Note that if the maximal element with minimal frequency multiset selector were used instead, only 2 permutations, namely $h_{1,1} \circ h_{2,1}$ and $h_{1,1} \circ h_{2,2}$, would be compatible with *pval*. ♣

The following property of compatibility is crucial in latter sections.

Theorem 2.5 *Let $g \in G$. A permutation $\hat{g} \in G$ is compatible with a place valuation pval if and only if the permutation $g \circ \hat{g} \in G$ is compatible with the permuted place valuation $g(pval)$.*

Furthermore, it is straightforward to see that if

- the place valuation *pval* is an injection, i.e., $pval(p) = pval(p') \Rightarrow p = p'$ for all places $p, p' \in P$, and
- *select* is a function multiset selector,

then there is exactly one element in G that is compatible with *pval*. Algorithm 2.2 describes the obvious depth-first backtrack search algorithm enumerating all permutations compatible with a place valuation.

3 USING THE CANONICAL VERSION OF THE CHARACTERISTIC GRAPH

Consider a P/T-net $N = \langle P, T, F, W, M_0 \rangle$ and the stabilizer group $G = \mathrm{Stab}(N, \hat{M})$ of a marking $\hat{M}$. Usually, $\hat{M}$ is either the initial marking M_0 or the null marking (in the latter case, $\mathrm{Stab}(N, \hat{M})$ equals to $\mathrm{Aut}(N)$). A *characteristic graph assigner* (under G) is a function that assigns each marking M a graph $\mathcal{G}_M$ (in a fixed class of graphs) such that its vertex set contains $P \cup T$ and for all markings M_1, M_2 of N it holds that

Algorithm 2.2 Enumerating all compatible permutations

function $compatible\_permutations(pval)$
  Call $backtrack(1, \mathbf{I})$
function $backtrack(l, \hat{g})$
Require: l is the backtracking level
Require: $\hat{g}$ is the currently enumerated compatible permutation
  if $l = |P| + 1$ **then**
    Report $\hat{g} \circ g'$ for each $g' \in U_{|P|+1} \circ \cdots \circ U_{|P|+|T|}$
    return
  evaluate $S = select(\Sigma_{h \in U_l} 1'pval(\hat{g}(h(p_{\beta,l}))))$
  for all $h \in U_l$ such that $pval(\hat{g}(h(p_{\beta,l}))) \in S$ **do**
    Call $backtrack(l + 1, \hat{g} \circ h)$
  return

1. if $g \in G$ maps a marking M_1 to M_2, then there is an isomorphism γ from $\mathcal{G}_{M_1}$ to $\mathcal{G}_{M_2}$ such that γ restricted to $P \cup T$ equals to g, and
2. if γ is an isomorphism from $\mathcal{G}_{M_1}$ to $\mathcal{G}_{M_2}$, then (i) $\gamma(P) = P$, (ii) $\gamma(T) = T$, and (iii) γ restricted to $P \cup T$ belongs to G and maps M_1 to M_2.

Then the graph $\mathcal{G}_M$ is called the *characteristic graph* of M. Clearly, two markings are symmetric under G if and only if their characteristic graphs are isomorphic. Thus testing whether two markings are symmetric under G can be done by (i) building their characteristic graphs, and (ii) testing whether the characteristic graphs are isomorphic by using a tool for solving the graph isomorphism problem. Furthermore, the stabilizer group $\mathrm{Stab}(G, M)$ can be easily retrieved from the automorphism group of $\mathcal{G}_M$ by simply restricting it to $P \cup T$.

Directed, vertex and edge labeled graphs. For this class of graphs, it is easy to define a characteristic graph assigner function. One can simply define that the characteristic graph of a marking M is the graph $\mathcal{G}_M = \langle V, E, L \rangle$ such that

1. the vertex set is the set of nodes of the net: $V = P \cup T$,
2. the edges are the arcs of the net N: $E = F$, and
3. each place $p \in P$ is labeled with the sequence of numbers defined by the markings $\hat{M}$ and M: $L(p) = \hat{M}(p).M(p)$
4. each transition $t \in T$ is labeled with the text string "T", $L(t) =$ "T", so that it is distinguished from the vertices representing the places, and
5. each edge $f \in F$ is labeled with the arc multiplicity $L(f) = W(f)$.

An isomorphism γ from a graph $G_1 = \langle V_1, E_1, L_1 \rangle$ to $G_2 = \langle V_2, E_2, L_2 \rangle$ is now a bijection from V_1 to V_2 such that

1. $\langle v, v' \rangle \in E_1 \Leftrightarrow \langle \gamma(v), \gamma(v') \rangle \in E_2$,
2. $L_1(v) = L_2(\gamma(v))$ for each vertex $v \in V_1$, and
3. $L_1(\langle v, v' \rangle) = L_2(\langle \gamma(v), \gamma(v') \rangle)$ for each edge $\langle v, v' \rangle \in E_1$.

It is straightforward to see that the requirements for a characteristic graph assigner are fulfilled by the above definition.

Undirected, vertex labeled graphs. For this class of graphs, some extra vertices and edges are inserted to compensate the lack of edge labels and direction. One can define that the characteristic graph of a marking M is the graph $\mathcal{G}_M = \langle V, E, L\rangle$ such that

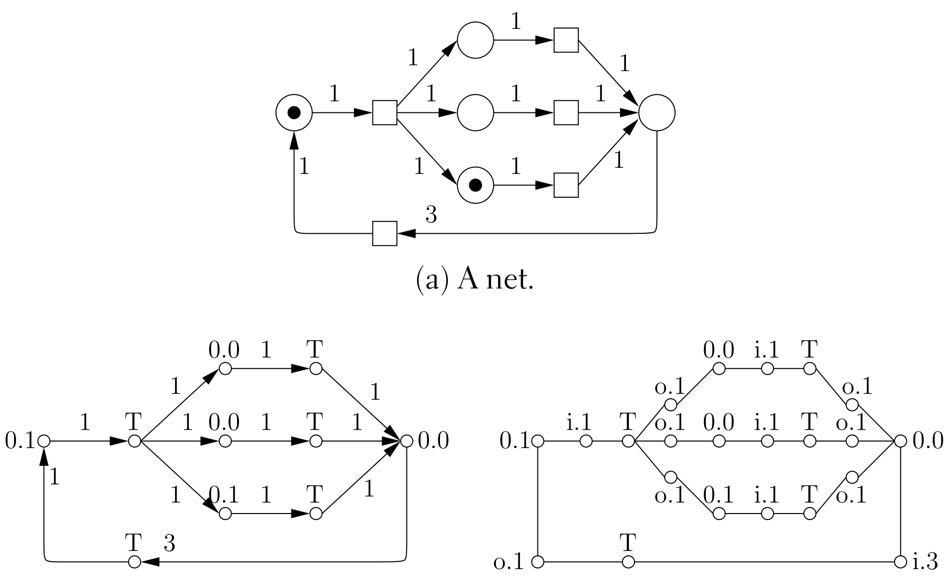
1. the vertex set is $V = P \cup T \cup F$,
2. for each arc $\langle x, y\rangle \in F$, the edge set E contains the edges $\langle x, \langle x, y\rangle\rangle$ and $\langle\langle x, y\rangle, y\rangle$, and these are the only edges in E,
3. for each place $p \in P$, $L(p) = \hat{M}(p).M(p)$,
4. for each transition $t \in T$, $L(t)$ is the text string "T", and
5. for each arc $\langle p, t\rangle \in F \cap (P \times T)$, $L(\langle p, t\rangle)$ is the concatenation of the text string "i" (for input arc) and the number $W(\langle p, t\rangle)$ and for each arc $\langle t, p\rangle \in F \cap (T \times P)$, $L(\langle t, p\rangle)$ is the concatenation of the text string "o" (for output arc) and the number $W(\langle t, p\rangle)$.

An isomorphism γ from a graph $G_1 = \langle V_1, E_1, L_1\rangle$ to $G_2 = \langle V_2, E_2, L_2\rangle$ is now a bijection from V_1 to V_2 such that

1. $\langle v, v'\rangle \in E_1 \Leftrightarrow \langle\gamma(v), \gamma(v')\rangle \in E_2$, and
2. $L_1(v) = L_2(\gamma(v))$ for each vertex $v \in V_1$.

It is again straightforward to see that the requirements for a characteristic graph assigner are fulfilled by the above definition.

Figure 6 shows a marked net and its characteristic graphs for both of the graph classes mentioned above (the marking $\hat{M}$ is assumed to be the empty marking).



(a) A net.

(b) Directed, vertex and edge labeled characteristic graph.

(c) Undirected, vertex labeled characteristic graph.

Figure 6: A marked net and its characteristic graphs.

The characteristic graph assigners defined above can be improved in the case the group $G = \mathrm{Stab}(N, \hat{M})$ is given. Assume that the set of nodes $P \cup T$ of the net is ordered. The orbits of the nodes under G, $[x]_G = \{g(x) \mid g \in G\}$ for each $x \in P \cup T$, inherit the same ordering by e.g. considering the first element in each orbit. Let *orbitnum*$(x) = i$ if the node $x \in P \cup T$ belongs to the ith orbit. Now the labels of the vertices in the characteristic graph corresponding to the places and transitions can be replaced by (i) $L(p) =$ *orbitnum*$(p).M(p)$ for each place p, and (ii) $L(t) =$ *orbitnum*(t)."T" for each transition t. Note that this construction requires that the group G is the stabilizer group of a marking, it does *not* work for arbitrary subgroups of $\mathrm{Aut}(N)$.

Graph canonizers. For a fixed class of graphs (closed under isomorphisms), a function $\mathcal{K}$ from graphs to graphs in the class is a *graph canonizer* if for all graphs G, G' it holds that

- $\mathcal{K}(G)$ is isomorphic to G, and
- $\mathcal{K}(G) = \mathcal{K}(G')$ if and only if G and G' are isomorphic.

The graph $\mathcal{K}(G)$ is the *canonical version* of G. It can be assumed that the vertex set of the canonical version of a graph with n vertices is $\{1, 2, \ldots, n\}$ and that a bijective *canonization mapping*, i.e., an isomorphism from G to $\mathcal{K}(G)$, is provided, too.

A graph canonizer can be used for obtaining canonical representative markings, as shown next. First, it is assumed that a Schreier-Sims representation for the group $G = \mathrm{Stab}(N, \hat{M})$ is given. For a marking $M \in \mathbb{M}$, consider the following procedure.

1. Build the characteristic graph $\mathcal{G}_M$.
2. Compute the canonical version $\mathcal{K}(\mathcal{G}_M)$ of $\mathcal{G}_M$ and a canonization mapping γ from $\mathcal{G}_M$ to $\mathcal{K}(\mathcal{G}_M)$.
3. Define the place valuation *pval* by $\forall p \in P$: *pval*$(p) = \gamma(p)$, i.e., the place p is associated with the number of the vertex into which the vertex p in the characteristic graph is mapped by γ. Clearly, *pval* is an injection.
4. Take the unique element $\hat{g} \in G$ that is compatible with *pval* (under a fixed function multiset selector *select*).
5. Return $\hat{g}^{-1}(M)$ as the representative marking.

Denote the marking $\hat{g}^{-1}(M)$ above by $\mathcal{K}_{\mathbb{M}}(M)$. The fact that $\mathcal{K}_{\mathbb{M}}(M)$ is unique for M despite the indefinite article at item 2 in the process described above (i.e., any canonization mapping can be selected) is proven in the following theorem.

Theorem 3.1 *The mapping $\mathcal{K}_{\mathbb{M}}$ is a canonical representative function.*

Example 3.2 Consider the marked version of the net N in Fig. 4, shown in the left hand side of Fig. 7. The characteristic graph $\mathcal{G}_M$ of the marking (when $\hat{M}$ is the empty marking) is shown in the middle of Fig. 7.

Suppose a graph canonizer that produces the canonical version $\mathcal{K}(\mathcal{G}_M)$ of $\mathcal{G}_M$ shown in the right hand side of Fig. 7. There are two isomorphisms, i.e., canonization mappings, from the characteristic graph $\mathcal{G}_M$ to its canonical version $\mathcal{K}(\mathcal{G}_M)$, namely

$$\begin{aligned}\gamma_1 &= \left(\begin{smallmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ 10 & 12 & 11 & 9 & 3 & 8 & 1 & 7 & 4 & 2 & 6 & 5 \end{smallmatrix}\right), \text{ and} \\ \gamma_2 &= \left(\begin{smallmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ 10 & 9 & 11 & 12 & 5 & 6 & 2 & 4 & 7 & 1 & 8 & 3 \end{smallmatrix}\right).\end{aligned}$$

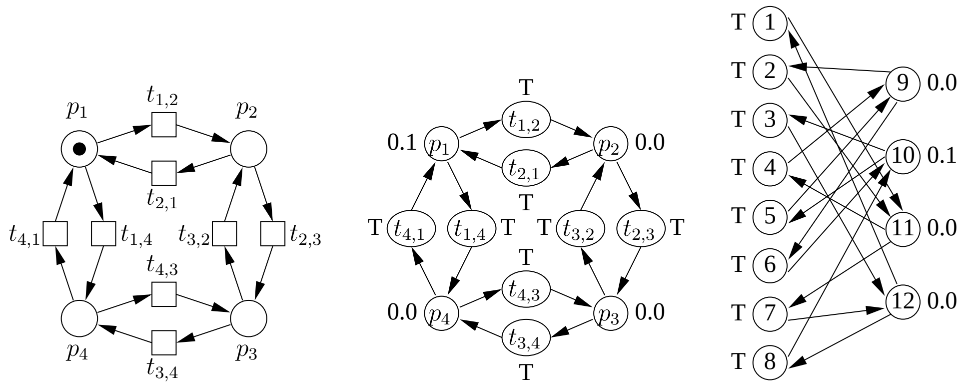


Figure 7: A marked net, its characteristic graph, and the canonical version of the characteristic graph

The corresponding place valuations are

$$\begin{aligned} \mathit{pval}_1 &= \{p_1 \mapsto 10, p_2 \mapsto 12, p_3 \mapsto 11, p_4 \mapsto 9\}, \text{ and} \\ \mathit{pval}_2 &= \{p_1 \mapsto 10, p_2 \mapsto 9, p_3 \mapsto 11, p_4 \mapsto 12\}, \end{aligned}$$

respectively. Assuming the Schreier-Sims representation of $\mathrm{Aut}(N)$ used in Ex. 2.2 and that the minimal element multiset selector is applied,

$$\hat{g}_1 = h_{1,4} \circ h_{2,1} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_4 & p_1 & p_2 & p_3 & t_{4,1} & \cdots \end{pmatrix}$$

is the only permutation compatible with pval_1 and

$$\hat{g}_2 = h_{1,2} \circ h_{2,2} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_2 & p_1 & p_4 & p_3 & t_{2,1} & \cdots \end{pmatrix}$$

is the only permutation compatible with pval_2. The canonical representative marking for M is thus

$$\hat{g}_1^{-1}(M) = \hat{g}_2^{-1}(M) = 1'p_2.$$

Finally, note that

- $\mathrm{Stab}(\mathrm{Aut}(N), M) = \{\mathbf{I}, h_{2,2}\}$,
- $\mathrm{Aut}(\mathcal{K}(\mathcal{G}_M)) = \{\mathbf{I}, \left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 2 & 1 & 5 & 7 & 3 & 8 & 4 & 6 & 12 & 10 & 11 & 9 \end{smallmatrix}\right)\}$,
- $\mathrm{Aut}(\mathcal{G}_M) = \gamma_1^{-1} \circ \mathrm{Aut}(\mathcal{K}(\mathcal{G}_M)) \circ \gamma_1 = \gamma_2^{-1} \circ \mathrm{Aut}(\mathcal{K}(\mathcal{G}_M)) \circ \gamma_2 = \left\{\mathbf{I}, \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_1 & p_4 & p_3 & p_2 & t_{1,4} & t_{4,1} & t_{4,3} & t_{3,4} & t_{3,2} & t_{2,3} & t_{2,1} & t_{1,2} \end{pmatrix}\right\}$, and
- $\mathrm{Stab}(\mathrm{Aut}(N), M)$ equals to $\mathrm{Aut}(\mathcal{G}_M)$ restricted to $P \cup T$ (i.e. $\mathrm{Aut}(\mathcal{G}_M)$ for the class of characteristic graphs used here).

♣

4 BACKTRACK SEARCH IN THE SCHREIER-SIMS REPRESENTATION

The algorithm presented in this section is based on selecting a permutation that is compatible with the marking in question. That is, *the marking itself is interpreted as a place valuation*. A canonical representative function is obtained by performing a backtracking search in the Schreier-Sims representation for the lexicographically smallest marking produced by a compatible permutation. Pruning techniques for the search are also discussed.

First, assume a base $\beta = [p_{\beta,1}, \ldots, p_{\beta,|P|}, t_{\beta,1}, \ldots, t_{\beta,|T|}]$ where the places are enumerated before the transitions and a Schreier-Sims representation $\vec{G} = [U_1, \ldots, U_{|P|+|T|}]$ of any subgroup G of $\mathrm{Aut}(N)$ under this base. Similarly, a fixed multiset selector is implicitly assumed throughout this and following sections.

Let

$$\mathit{posreps}(M) = \{\hat{g}^{-1}(M) \mid \hat{g} \in G \text{ and } \hat{g} \text{ is compatible with } M\}$$

denote the set of *possible representative markings* for M. For symmetric markings, the sets of possible representative markings are the same:

Theorem 4.1 *For each marking $M \in \mathbb{M}$ and for each symmetry $g \in G$, $\mathit{posreps}(M) = \mathit{posreps}(g(M))$.*

Obviously, $M' \in \mathit{posreps}(M)$ implies $M' \equiv_G M$. However, it does *not*, in general, hold that $M \in \mathit{posreps}(M)$. Note that the number of symmetries in G compatible with M is a multiple of $|\mathrm{Stab}(G, M)|$: if $\hat{g}$ is compatible with M, then by Thm. 2.5 the permutation $g \circ \hat{g} \in G$ is compatible with the marking $g(M) = M$ for each stabilizer $g \in \mathrm{Stab}(G, M)$. That is, if $\hat{g}$ is compatible with M, then (and only then) all the permutations in the right coset $\mathrm{Stab}(G, M) \circ \hat{g}$ are compatible with M.

The "hardness" of a marking can be classified as follows. Define that a marking M is

1. *trivial* if there is exactly one permutation in G compatible with the marking M,
2. *easy* if it is not trivial but the set $\mathit{posreps}(M)$ contains only one marking,
3. *hard* if it is neither trivial nor easy.

Note that this classification depends on the applied Schreier-Sims representation and multiset selector. It is easy to see that the classification is closed under G, i.e. a marking is trivial/easy/hard if and only if all the markings symmetric to it under G are trivial/easy/hard, respectively. Note that for both trivial and easy markings, the set $\mathit{posreps}(M)$ contains only one marking. The difference is that easy markings have several permutations in G that are compatible with the marking.

A very simple representative marking algorithm would be to simply take an arbitrary permutation $\hat{g}$ that is compatible with the marking M in question and then return $\hat{g}^{-1}(M)$ as the representative marking. Theorem 4.1 guarantees that it is possible, although not guaranteed, that the same representative marking is selected for two symmetric markings. However, for trivial and easy markings, as classified above, the unique canonical representative marking is returned.

A canonical representative marking algorithm can be obtained by first defining a total order between all the markings and then selecting the smallest (or greatest) marking in the set of possible representative markings to be the representative marking. A natural total ordering between the markings is the lexicographical ordering. Formally, a marking M_1 is lexicographically smaller than a marking M_2 under the base β, denoted by $M_1 <_\beta M_2$, if there is a number $1 \leq k \leq |P|$ such that $M_1(p_{\beta,j}) = M_2(p_{\beta,j})$ for all $1 \leq j < k$ and $M_1(p_{\beta,k}) < M_2(p_{\beta,k})$. Define that $M_1 \leq_\beta M_2$ if $M_1 = M_2$ or $M_1 <_\beta M_2$. Now define *canrepr*(M) to be the $<_\beta$-smallest marking in *posreps*(M). As *posreps*(M) = *posreps*$(g(M))$, *canrepr*(M) = *canrepr*$(g(M))$ for all markings M and for all $g \in G$. Furthermore, *canrepr*$(M) \equiv_G M$. The canonical representative marking *canrepr*(M) for a marking M can be obtained by the backtracking depth-first search algorithm Alg. 4.1 derived from Alg. 2.2.

Algorithm 4.1 A function returning the lexicographically smallest marking in the set *posreps*(M)

---

function *canrepr*(M)
Require: A global marking $BestMarking$
Set $BestMarking = p \mapsto \infty$ for all $p \in P$
Set *pval*$(p) = M(p)$ for each place p
Call $backtrack(1, \mathbf{I})$
return $BestMarking$
function $backtrack(l, \hat{g})$
Require: l is the backtracking level
Require: $\hat{g}$ is the currently enumerated compatible permutation
if $l = |P| + 1$ **then**
  if $\hat{g}^{-1}(M) \leq_\beta BestMarking$ **then**
    Set $BestMarking = \hat{g}^{-1}(M)$
  return
evaluate $S = select(\Sigma_{h \in U_l} 1'pval(\hat{g}(h(p_{\beta,l}))))$
for all $h \in U_l$ such that $pval(\hat{g}(h(p_{\beta,l}))) \in S$ **do**
  Call $backtrack(l + 1, \hat{g} \circ h)$
return

Example 4.2 Recall the net in Fig. 4 and the Schreier-Sims representation of its automorphism group G described in Ex. 2.2. In Ex. 2.4, it was shown that the symmetries

$$\begin{aligned}
h_{1,2} \circ h_{2,1} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \dots \\ p_2 & p_3 & p_4 & p_1 & t_{2,3} & \dots \end{pmatrix}, \\
h_{1,3} \circ h_{2,1} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \dots \\ p_3 & p_4 & p_1 & p_2 & t_{3,4} & \dots \end{pmatrix}, \\
h_{1,3} \circ h_{2,2} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \dots \\ p_3 & p_2 & p_1 & p_4 & t_{3,2} & \dots \end{pmatrix}, \text{ and} \\
h_{1,4} \circ h_{2,2} &= \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \dots \\ p_4 & p_3 & p_2 & p_1 & t_{4,3} & \dots \end{pmatrix}
\end{aligned}$$

are compatible with the marking $M = 1'p_1$ (under the minimal element multiset selector). Thus *posreps*$(M) = \{1'p_3, 1'p_4\}$ and M is hard. Under the applied base, $1'p_4$ is the lexicographically smallest marking in the set *posreps*(M). ♣

Pruning with the already fixed prefix. Consider a permutation $g = g_1 \circ \cdots \circ g_i$ in G, where $1 \leq i \leq |P|$ and $g_j \in U_j$ for each $1 \leq j \leq i$. Now each "extended" permutation $\tilde{g} = g_1 \circ \cdots \circ g_i \circ g_{i+1} \circ g_{|P|+|T|}$ in G maps $p_{\beta,1}$ to $g(p_{\beta,1})$, $p_{\beta,2}$ to $g(p_{\beta,2})$, and so on up to and including $p_{\beta,i}$ that is mapped to $g(p_{\beta,i})$. Thus the values of the first i places in $\tilde{g}^{-1}(M)$ are known: $(\tilde{g}^{-1}(M))\,(p_{\beta,1}) = M(\tilde{g}(p_{\beta,1})) = M(g(p_{\beta,1}))$, …, and $(\tilde{g}^{-1}(M))\,(p_{\beta,i}) = M(\tilde{g}(p_{\beta,i})) = M(g(p_{\beta,i}))$. If a marking $M' \in$ *posreps*(M) such that (i) $M'(p_{\beta,j}) = M(g(p_{\beta,j}))$ for each $1 \leq j < k$ and (ii) $M'(p_{\beta,k}) < M(g(p_{\beta,k}))$ for a $1 \leq k \leq i$ has already been found during the search, one knows that $M' <_\beta \tilde{g}^{-1}(M)$ for all extensions $\tilde{g}$ of g and can therefore skip all such $\tilde{g}$.

To improve the possibilities of this pruning technique to work efficiently, the Schreier-Sims representation can be optimized to have the fixed elements as early as possible in the base. Let $p_{\beta,i}$ be the last element in the base where a place $p_{\beta,j}$, $j \geq i$, may be permuted i.e. $h_{i,l}(p_{\beta,j}) \neq p_{\beta,j}$ for an $h_{i,l} \in U_i$. Now the base can be changed so that $p_{\beta,j}$ is after $p_{\beta,i}$ but before any $p_{\beta,k}$ for which $U_k \supset \{\mathbf{I}\}$.

Finding and pruning with stabilizers. Take any "prefix" permutation $\tilde{g} = g_1 \circ \cdots \circ g_{i-1} \in U_1 \circ \cdots \circ U_{i-1}$ for an $1 \leq i \leq |P|$. Consider two left cosets, $(\tilde{g} \circ g_i) \circ G_{i+1}$ and $(\tilde{g} \circ g'_i) \circ G_{i+1}$, where $g_i, g'_i \in U_i$. Let π be a stabilizer of a marking M that (i) fixes each place $\tilde{g}(p_{\beta,1}), \ldots, \tilde{g}(p_{\beta,i-1})$, and (ii) maps the place $(\tilde{g} \circ g_i)(p_{\beta,i})$ to $(\tilde{g} \circ g'_i)(p_{\beta,i})$. Now, if a permutation g' belongs to the left coset $(\tilde{g} \circ g'_i) \circ G_{i+1}$, then $\pi^{-1} \circ g'$ must belong to the left coset $(\tilde{g} \circ g_i) \circ G_{i+1}$ since (i) $(\pi^{-1} \circ g')(p_{\beta,j}) = \pi^{-1}(\tilde{g}(p_{\beta,j})) = \tilde{g}(p_{\beta,j})$ for each $1 \leq j < i$ and (ii) $(\pi^{-1} \circ g')(p_{\beta,i}) = \pi^{-1}((\tilde{g} \circ g'_i)(p_{\beta,i})) = (\tilde{g} \circ g_i)(p_{\beta,i})$. Furthermore, for each marking M, $(\pi^{-1} \circ g')^{-1}(M) = (g'^{-1} \circ \pi)(M) = g'^{-1}(M)$. Therefore, the left cosets $(\tilde{g} \circ g'_i) \circ G_{i+1}$ and $(\tilde{g} \circ g_i) \circ G_{i+1}$ produce the same markings. In addition, if g is compatible with M, then $\pi^{-1} \circ g$ is compatible with $\pi^{-1}(M) = M$ and therefore the sets of possible representative markings in the left cosets are the same. To sum up, if all the permutations in a left coset $(\tilde{g} \circ g_i) \circ G_{i+1}$ have already been searched and there is a stabilizer π with the above mentioned properties, one can ignore the left coset $(\tilde{g} \circ g'_i) \circ G_{i+1}$ as it produces the same possible representative markings.

Stabilizers of markings can be found during the backtrack search on the Schreier-Sims representation. Consider that M' is a marking that has been found earlier during the search by traversing a path $g = g_1 \circ \cdots \circ g_{i-1} \circ g_i \circ g_{i+1} \circ \cdots g_{|P|}$ meaning that $g^{-1}(M) = M'$. For instance, M' could be the lexicographically smallest marking found so far. Assume that the currently traversed path is $g' = g_1 \circ \cdots \circ g_{i-1} \circ g'_i \circ g'_{i+1} \circ \cdots g'_{|P|}$, where $g'_i \neq g_i$. If it holds that $g'^{-1}(M) = M' = g^{-1}(M)$, then $g' \circ g^{-1}$ is a stabilizer of M and (i) $g' \circ g^{-1}$ fixes each $(g_1 \circ \cdots \circ g_j)(p_{\beta,j})$, $1 \leq j < i$, as $(g' \circ g^{-1})((g_1 \circ \cdots \circ g_j)(p_{\beta,j})) = (g' \circ g^{-1})(g(p_{\beta,j})) = g'(p_{\beta,j}) = (g_1 \circ \cdots \circ g_j)(p_{\beta,j})$, and (ii) $g' \circ g^{-1}$ maps $(g_1 \circ \cdots \circ g_{i-1} \circ g_i)(p_{\beta,i}) = g(p_{\beta,i})$ to $g'(p_{\beta,i}) = (g_1 \circ \cdots \circ g_{i-1} \circ g'_i)(p_{\beta,i})$. Thus $g' \circ g^{-1}$ is a stabilizer of M fulfilling the properties discussed above (the prefix $\tilde{g}$ being $g_1 \circ \cdots \circ g_{i-1}$), and the search can be "back-jumped" to the level $i - 1$. This is the most trivial (and easiest to implement) way to prune with the found stabilizers. There are many ways to achieve even larger degree of pruning by composing the found stabilizers, see [Kreher and Stinson 1999; McKay 1981; Butler 1991].

Transition pruning with the stabilizers. Stabilizers of markings can also be used to prune the set of successor markings that has to be visited during the reduced reachability graph generation, see e.g. [Jensen 1995]. Namely, if $g \in \mathrm{Stab}(G, M)$, then $M\ [t\rangle\ M' \Leftrightarrow M\ [g(t)\rangle\ g(M')$. That is, firing transitions that are in the same orbit under $\mathrm{Stab}(G, M)$ lead to symmetric successor markings. As generators of $\mathrm{Stab}(G, M)$ are found during the search through the Schreier-Sims representation, the orbits of the transitions under $\mathrm{Stab}(G, M)$ can be computed during the search, too.

5 PARTITION GUIDED SCHREIER-SIMS SEARCH

It is possible to combine the backtracking search in the Schreier-Sims representation described in Sect. 4 with a standard preprocessing technique applied in graph isomorphism algorithms. Assuming a fixed subgroup G of $\mathrm{Aut}(N)$ and given a marking M, an ordered partition of $P \cup T$ is first computed in a way that respects the symmetries in G. The procedure computing the partition for M is based on the use of invariants and is a variant of the standard algorithms for graph isomorphism checking and canonical labeling of graphs, see e.g. [McKay 1981; Kreher and Stinson 1999]. The place valuation corresponding to the partition is then used to prune the search in the Schreier-Sims representation of G. That is, instead of searching through the permutations that are compatible with the marking in question as was done in Sect. 4, the permutations compatible with the constructed place valuation are searched. The hope is that the place valuation is closer to being injective than the original marking, i.e., that it can distinguish the places from each other better.

5.1 Ordered Partitions

Some notation and preliminaries of ordered partitions is defined first.

An *ordered partition* of a nonempty set A is a list $\mathfrak{p} = [C_1, \ldots, C_n]$ such that the set $\{C_1, \ldots, C_n\}$ is a partition of A meaning that (i) $\emptyset \neq C_i \subseteq A$ for each $1 \leq i \leq n$, (ii) $\bigcup_{i=1}^{n} C_i = A$, and (iii) $C_i \cap C_j = \emptyset$ for all $i \neq j$. The sets C_i are the *cells* (or blocks) of the partition. An ordered partition is *discrete* if all its cells are singleton sets and *unit* if it contains only one cell (the set A). The function *incell* from the ordered partitions of A and the elements of A to natural numbers is defined by $\mathit{incell}([C_1, \ldots, C_n], x) = i \Leftrightarrow x \in C_i$.

An ordered partition $\mathfrak{p}_1$ of A is *finer than* (or a *refinement of*) an ordered partition $\mathfrak{p}_2$, denoted by $\mathfrak{p}_1 \leq \mathfrak{p}_2$, if each cell in $\mathfrak{p}_1$ is a subset of a cell in $\mathfrak{p}_2$. An ordered partition $\mathfrak{p}_1$ of A is a *cell order preserving refinement* of an ordered partition $\mathfrak{p}_2$, denoted by $\mathfrak{p}_1 \preceq \mathfrak{p}_2$, if $\mathfrak{p}_1 \leq \mathfrak{p}_2$ and for all $x, y \in A$, $\mathit{incell}(\mathfrak{p}_1, x) < \mathit{incell}(\mathfrak{p}_1, y)$ implies $\mathit{incell}(\mathfrak{p}_2, x) \leq \mathit{incell}(\mathfrak{p}_2, y)$. That is, if $\mathfrak{p}_2 = [C_{2,1}, \ldots, C_{2,n}]$, then any $\mathfrak{p}_1$ such that $\mathfrak{p}_1 \preceq \mathfrak{p}_2$ is of form $[C_{1,1,1}, \ldots, C_{1,1,d_1}, \ldots C_{1,n,1}, \ldots, C_{1,n,d_n}]$, where $\bigcup_{1 \leq j \leq d_i} C_{1,i,j} = C_{2,i}$ for all $1 \leq i \leq n$. For instance, $[\{b\}, \{a\}, \{c\}] \leq [\{a\}, \{b, c\}]$, $[\{b\}, \{a\}, \{c\}] \not\preceq [\{a\}, \{b, c\}]$, and $[\{a\}, \{c\}, \{b\}] \preceq [\{a\}, \{b, c\}]$. The relation $\preceq$ is reflexive, transitive and antisymmetric, i.e., a partial order on the set of all ordered partitions of A.

A permutation γ of A acts on ordered partitions of A by $\gamma([C_1, \ldots, C_n]) = [\gamma(C_1), \ldots, \gamma(C_n)]$. Clearly, $incell(\mathfrak{p}, x) = incell(\gamma(\mathfrak{p}), \gamma(x))$ for all ordered partitions $\mathfrak{p}$ of A and for all $x \in A$.

5.2 Partition Generators

Assume a net N, a subgroup G of $\mathrm{Aut}(N)$, and a Schreier-Sims representation $\vec{G} = [U_1, \ldots, U_{|P|+|T|}]$ of G under a base $\beta = [p_1, \ldots, p_{|P|}, t_1, \ldots, t_{|T|}]$ in which the places are enumerated before the transitions. Denote the set of all ordered partitions of $P \cup T$ by $\mathfrak{P}$.

Next, the marking M in question is assigned an ordered partition of $P \cup T$ in a way that respects the symmetries in G. The idea is to try to distinguish between the elements in $P \cup T$ so that distinguishable elements are put in different cells. Formally, define the following.

Definition 5.1 *A function $f : \mathbb{M} \to \mathfrak{P}$ assigning each marking to an ordered partition is a G-*partition generator *if for all markings $M \in \mathbb{M}$ and for all $g \in G$ it holds that $f(g(M)) = g(f(M))$.*

That is, for permuted markings, similarly permuted ordered partitions are assigned. A technique for obtaining G-partition generators is described in Sect. 5.3. Now assume a fixed G-partition generator f.

An ordered partition can be interpreted as a place valuation by simply assigning each place the cell number in which it appears in the ordered partition. Formally, the place valuation $pval_{\mathfrak{p}}$ corresponding to an ordered partition $\mathfrak{p}$ of $P \cup T$ is defined by

$$pval_{\mathfrak{p}}(p) = incell(\mathfrak{p}, p)$$

for each place $p \in P$. The next lemma shows that the place valuations assigned to symmetric markings in this way are symmetric, too.

Lemma 5.2 *For all $g \in G$ and all markings M, $pval_{f(g(M))} = g(pval_{f(M)})$.*

A direct consequence of this is that each stabilizer $g \in \mathrm{Stab}(G, M)$ is a stabilizer of $pval_{f(M)}$:

Corollary 5.3 *For each stabilizer $g \in \mathrm{Stab}(G, M)$, $g(pval_{f(M)}) = pval_{f(M)}$.*

Thus $\mathrm{Stab}(G, M)$ is a subgroup of $\mathrm{Stab}(G, pval_{f(M)})$. For all "reasonable" G-partition generators, the stabilizer groups are actually the same.[2]

Lemma 5.4 *If $incell(f(M), p_1) = incell(f(M), p_2) \Rightarrow M(p_1) = M(p_2)$ for all places $p_1, p_2 \in P$, then $\mathrm{Stab}(G, pval_{f(M)}) = \mathrm{Stab}(G, M)$.*

After building the ordered partition $f(M)$ for the marking M and the corresponding place valuation $pval_{f(M)}$, let

$$posreps(M) = \{\hat{g}^{-1}(M) \mid \hat{g} \in G \text{ and } \hat{g} \text{ is compatible with } pval_{f(M)}\}$$

[2] Such "reasonable" cases are obtained by simply applying the marking invariant described in the following subsection during the partition generation process.

denote the set of *possible representative markings* for M (recall Sect. 4). Like earlier in Thm. 4.1, it can be proven that for symmetric markings, the sets of possible representative markings coincide.

Theorem 5.5 *For each marking $M \in \mathbb{M}$ and for each symmetry $g \in G$, $posreps(M) = posreps(g(M))$.*

Again, $M' \in posreps(M)$ implies $M' \equiv_G M$ and it is *not*, in general, the case that $M \in posreps(M)$. Furthermore, by Thm. 2.5, a permutation $\hat{g}$ is compatible with $pval_{f(M)}$ if and only if the permutation $g \circ \hat{g}$ is compatible with $g(pval_{f(M)}) = pval_{f(M)}$ for any stabilizer $g \in G$ of $pval_{f(M)}$. Hence the number of permutations compatible with $pval_{f(M)}$ is a multiple of $\left|\mathrm{Stab}(G, pval_{f(M)})\right|$ (that in all reasonable cases equals to $|\mathrm{Stab}(G, M)|$ by Lemma 5.4).

Now the lexicographically smallest state in $posreps(M)$ can be searched by using the backtrack search algorithm Alg. 4.1 described in Sect. 4 with the obvious changes (i.e., changing the line 2 to refer to the ordered partition $f(M)$). Obviously, the pruning technique based on the fixed prefix is sound, and Cor. 5.3 ensures that the stabilizer pruning technique is also sound.

Similarly to that in Sect. 4, a hardness measure can be defined for markings. Define that a marking M is

1. *trivial* if the partition $f(M)$ is discrete,
2. *easy* if it is not trivial but the set $posreps(M)$ contains only one marking,
3. *hard* if it is neither trivial nor easy.

Again, this classification depends on the applied (i) Schreier-Sims representation, (ii) G-partition generator, and (iii) multiset selector. Furthermore, the classification is closed under G. Note that if a marking M is trivial, then there is a unique permutation in G compatible with the partition $f(M)$ and thus the set $posreps(M)$ contains only one marking. On the other hand, easy markings may have several permutations in G that are compatible with the partition. The definition of triviality defined here is stronger than that in Sect. 4: there may be markings M for which there is only one permutation compatible with $f(M)$ although $f(M)$ is not discrete. The definition here is chosen because it reveals the efficiency of the applied G-partition generator better (more trivial markings, the better). However, a fundamental limitation of G-partition generators is that they cannot distinguish between the elements that are in the same $\mathrm{Stab}(G, M)$-orbit:

Fact 5.6 *If $g \in \mathrm{Stab}(G, M)$ for a marking M, then $f(g(M)) = g(f(M))$ implies $f(M) = g(f(M))$ and thus each element $x \in P \cup T$ must be in the same cell in the partition $f(M)$ as the element $g(x)$.*

Thus a trivial marking M has the trivial stabilizer group, i.e., $\mathrm{Stab}(G, M) = \{\mathbf{I}\}$.

Example 5.7 Consider the net in Fig. 4 and the Schreier-Sims representation $\vec{G}$ of its automorphism group G described in Ex. 2.2.

Assume a marking $M = 1'p_1$ and a G-partition generator f mapping M to

$$f(M) = [\{p_3\}, \{p_2, p_4\}, \{p_1\}, \{t_{1,2}, t_{1,4}\}, \{t_{2,1}, t_{4,1}\}, \{t_{3,2}, t_{3,4}\}, \{t_{2,3}, t_{4,3}\}].$$

By Fact 5.6, this is one of the finest partitions that any G-partition generator can produce since $h_{1,1} \circ h_{2,2} = \left(\begin{smallmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_1 & p_4 & p_3 & p_2 & t_{1,4} & t_{4,1} & t_{4,3} & t_{3,4} & t_{3,2} & t_{2,3} & t_{2,1} & t_{1,2} \end{smallmatrix}\right)$ is a stabilizer of M in G. The corresponding place valuation is

$$pval_{f(M)} = \{p_1 \mapsto 3, p_2 \mapsto 2, p_3 \mapsto 1, p_4 \mapsto 2\}$$

and the symmetries in G compatible with $pval_{f(M)}$ are $h_{1,3} \circ h_{2,1}$ and $h_{1,3} \circ h_{2,2}$. Now $(h_{1,3} \circ h_{2,1})^{-1}(M) = 1'p_3$ and $(h_{1,3} \circ h_{2,2})^{-1}(M) = 1'p_3$. Thus $posreps(M) = \{1'p_3\}$. According to the above hardness measure for markings, M is easy.

For the marking $M' = 1'p_2$ (which is symmetric to M as $g = h_{1,2} \circ h_{2,1} = \left(\begin{smallmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_2 & p_3 & p_4 & p_1 & t_{2,3} & \cdots \end{smallmatrix}\right)$ maps M to M'), the G-partition generator f must map M' to $f(M') = f(g(M)) = g(f(M))$, i.e.,

$$f(M') = [\{p_4\}, \{p_1, p_3\}, \{p_2\}, \{t_{2,3}, t_{2,1}\}, \{t_{3,2}, t_{1,2}\}, \{t_{4,3}, t_{4,1}\}, \{t_{3,4}, t_{1,4}\}].$$

Again, this is one of the finest partitions one can get by using any G-partition generator since $h_{1,3} \circ h_{2,2} = \left(\begin{smallmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & \cdots \\ p_3 & p_2 & p_1 & p_4 & t_{3,2} & \cdots \end{smallmatrix}\right)$ is a stabilizer of M' in G. The corresponding place valuation is $pval_{f(M')} = \{p_1 \mapsto 2, p_2 \mapsto 3, p_3 \mapsto 2, p_4 \mapsto 1\}$ and the symmetries compatible with $pval_{f(M')}$ are $h_{1,4} \circ h_{2,1}$ and $h_{1,4} \circ h_{2,2}$. Now $(h_{1,4} \circ h_{2,1})^{-1}(M') = 1'p_3$ and $(h_{1,4} \circ h_{2,2})^{-1}(M') = 1'p_3$. Thus $posreps(M') = posreps(M)$ as required by Thm. 5.5. ♣

5.3 Partition Refiners and Invariants

The G-partition generators discussed above can be obtained by using G-partition refiners defined below.

Definition 5.8 *A G-partition refiner is a function $\mathcal{R} : \mathbb{M} \times \mathfrak{P} \to \mathfrak{P}$ such that both*

1. *$\mathcal{R}(M, \mathfrak{p}) \preceq \mathfrak{p}$, and*
2. *$\mathcal{R}(g(M), g(\mathfrak{p})) = g(\mathcal{R}(M, \mathfrak{p}))$*

hold for all $g \in G$, for all markings $M \in \mathbb{M}$, and for all partitions $\mathfrak{p} \in \mathfrak{P}$.

That is, the refined partition must be a cell order preserving refinement of the argument partition and for permuted arguments, the result has to be similarly permuted. A direct consequence of the definition is that if a permutation $g \in G$ fixes both a marking M and a partition $\mathfrak{p}$ (i.e., $G(M) = M$ and $g(\mathfrak{p}) = \mathfrak{p}$), then it fixes the refined partition $\mathcal{R}(M, \mathfrak{p})$, too. The composition $\mathcal{R}_2 \star \mathcal{R}_1$ of two G-partition refiners, $\mathcal{R}_1$ and $\mathcal{R}_2$, defined by $(\mathcal{R}_2 \star \mathcal{R}_1)(M, \mathfrak{p}) = \mathcal{R}_2(M, \mathcal{R}_1(M, \mathfrak{p}))$, is also a G-partition refiner. When a G-partition refiner is applied to the unit partition, the result is a G-partition generator.

Lemma 5.9 *For each G-partition refiner $\mathcal{R}$, the function $f_{\mathcal{R}} : \mathbb{M} \to \mathfrak{P}$ defined by $f_{\mathcal{R}}(M) = \mathcal{R}(M, [P \cup T])$ is a G-partition generator.*

A way to obtain G-partition refiners is based on the use of G-invariants.

Definition 5.10 *A function* $\mathcal{I} : \mathbb{M} \times \mathfrak{P} \times \{P \cup T\} \to \mathbb{Z}$ *is a* G-invariant *if*

$$\mathcal{I}(M, \mathfrak{p}, x) = \mathcal{I}(g(M), g(\mathfrak{p}), g(x)).$$

holds for all $g \in G$*, for all markings* $M \in \mathbb{M}$*, for all ordered partitions* $\mathfrak{p} \in \mathfrak{P}$ *of* $P \cup T$*, and for all nodes* $x \in P \cup T$*.*

Clearly any G-invariant is also a G'-invariant for any subgroup G' of G. The following are G-invariants for any subgroup G of $\mathrm{Aut}(N)$.

- The *node type invariant* $\mathcal{I}_{\text{node type}}$ is defined by

$$\mathcal{I}_{\text{node type}}(M, \mathfrak{p}, x) = \begin{cases} 0 & \text{if } x \in P \\ 1 & \text{if } x \in T. \end{cases}$$

- Assume a fixed total order between the places and transitions. Now the orbits of G inherit this order and the *G-orbit invariant* $\mathcal{I}_{G\text{-orbit}}$ is defined by $\mathcal{I}_{G\text{-orbit}}(M, \mathfrak{p}, x) = \mathit{orbitnum}(x)$, where $\mathit{orbitnum}(x)$ is defined as on page 14.
- The *marking invariant* $\mathcal{I}_{\text{marking}}$ is defined by

$$\mathcal{I}_{\text{marking}}(M, \mathfrak{p}, x) = \begin{cases} M(x) & \text{if } x \in P \\ -1 & \text{if } x \in T. \end{cases}$$

- The *preset* of an element $x \in P \cup T$ is the set ${}^{\bullet}x = \{x' \mid \langle x', x\rangle \in F\}$ and the *postset* $x^{\bullet}$ is the set $\{x' \mid \langle x, x'\rangle \in F\}$. The *partition independent weighted in- and out-degree invariants* are defined by

$$\mathcal{I}_{\text{in-degree of weight } w}(M, \mathfrak{p}, x) = |\{x' \in {}^{\bullet}x \mid W(\langle x', x\rangle) = w\}|$$

  and

$$\mathcal{I}_{\text{out-degree of weight } w}(M, \mathfrak{p}, x) = |\{x' \in x^{\bullet} \mid W(\langle x, x'\rangle) = w\}|\,.$$

- The *partition dependent weighted in- and out-degree invariants* are defined by

$$\mathcal{I}_{\text{in-degree of weight } w \text{ from cell } c}(M, \mathfrak{p}, x) = \\ |\{x' \in {}^{\bullet}x \mid W(\langle x', x\rangle) = w \wedge \mathit{incell}(\mathfrak{p}, x') = c\}|$$

  and

$$\mathcal{I}_{\text{out-degree of weight } w \text{ to cell } c}(M, \mathfrak{p}, x) = \\ |\{x' \in x^{\bullet} \mid W(\langle x, x'\rangle) = w \wedge \mathit{incell}(\mathfrak{p}, x') = c\}|\,.$$

Note that the partition independent weighted in- and out-degree invariants and the node type invariant are subsumed by the G-orbit invariant in the sense that if the values of two nodes are equal under the G-orbit invariant, they are equal under these invariants, too. That is, they cannot distinguish elements that the G-orbit invariant cannot.

A partition can be *refined according to an invariant* by splitting the cells according to the values assigned to nodes by the invariant. Formally, an invariant defines the corresponding partition refiner as follows. For a G-invariant $\mathcal{I}$, define the function $\mathcal{R}_{\mathcal{I}} : \mathbb{M} \times \mathfrak{P} \to \mathfrak{P}$ by $\mathcal{R}_{\mathcal{I}}(M, \mathfrak{p}) = \mathfrak{p}_r$ such that for all $x, x' \in \{P \cup T\}$, for all $\mathfrak{p} \in \mathfrak{P}$, and for all $M \in \mathbb{M}$,

1. $incell(\mathfrak{p}_r, x) = incell(\mathfrak{p}_r, x')$ if and only if $incell(\mathfrak{p}, x) = incell(\mathfrak{p}, x')$ and $\mathcal{I}(M, \mathfrak{p}, x) = \mathcal{I}(M, \mathfrak{p}, x')$, and
2. $incell(\mathfrak{p}_r, x) < incell(\mathfrak{p}_r, x')$ if and only if either
   (a) $incell(\mathfrak{p}, x) < incell(\mathfrak{p}, x')$, or
   (b) $incell(\mathfrak{p}, x) = incell(\mathfrak{p}, x')$ and $\mathcal{I}(M, \mathfrak{p}, x) < \mathcal{I}(M, \mathfrak{p}, x')$.

Lemma 5.11 *The function $\mathcal{R}_\mathcal{I}$ is a G-partition refiner.*

Partition refiners with respect to some invariants can also be defined procedurally so that in the resulting partition two nodes are in the same cell if and only if their invariant values in that partition are the same. This is especially the case for the partition dependent weighted in- and out-degree invariants, where the procedure corresponds to the method of computing the so-called equitable partition in [McKay 1981; Kreher and Stinson 1999].

Example 5.12 Consider again the net in Fig. 4 and the Schreier-Sims representation $\vec{G}$ of its automorphism group G described in Ex. 2.2.

Assume a marking $M = 1'p_1$. Initially, the partition is

$$\mathfrak{p}_{M,0} = [\{p_1, p_2, p_3, p_4, t_{1,2}, \ldots\}].$$

Refining this partition according to the G-orbit invariant yields

$$\mathfrak{p}_{M,1} = [\{p_1, p_2, p_3, p_4\}, \{t_{1,2}, \ldots\}],$$

and refining according to the marking M gives

$$\mathfrak{p}_{M,2} = [\{p_2, p_3, p_4\}, \{p_1\}, \{t_{1,2}, \ldots\}].$$

Evaluating the invariant $\mathcal{I}_{\text{in-degree of weight 1 from cell 1}}$ in the partition $\mathfrak{p}_{M,2}$ gives $\mathcal{I}_{\text{in-degree of weight 1 from cell 1}}(M, \mathfrak{p}_{M,2}, p_i) = 0$ for each $1 \le i \le 4$, and that $\mathcal{I}_{\text{in-degree of weight 1 from cell 1}}(M, \mathfrak{p}_{M,2}, t)$ equals to 0 for $t = t_{1,2}$ and $t = t_{1,4}$ and to 1 for other transitions. Refining $\mathfrak{p}_{M,2}$ thus yields

$$\mathfrak{p}_{M,3} = [\{p_2, p_3, p_4\}, \{p_1\}, \{t_{1,2}, t_{1,4}\}, \{t_{2,1}, t_{2,3}, t_{3,2}, t_{4,3}, t_{3,4}, t_{4,1}\}]$$

Refining this ordered partition according to $\mathcal{I}_{\text{in-degree of weight 1 from cell 2}}$ changes nothing and thus $\mathfrak{p}_{M,4} = \mathfrak{p}_{M,4}$. Next, $\mathcal{I}_{\text{in-degree of weight 1 from cell 3}}(M, \mathfrak{p}_{M,4}, p)$ equals to 0 for $p = p_1$ and $p = p_3$ and to 1 for $p = p_2$ and $p = p_4$, and $\mathcal{I}_{\text{in-degree of weight 1 from cell 3}}(M, \mathfrak{p}_{M,4}, t) = 0$ for all transitions. Thus

$$\mathfrak{p}_{M,5} = [\{p_3\}, \{p_2, p_4\}, \{p_1\}, \{t_{1,2}, t_{1,4}\}, \{t_{2,1}, t_{2,3}, t_{3,2}, t_{4,3}, t_{3,4}, t_{4,1}\}]$$

Refining with $\mathcal{I}_{\text{in-degree of weight 1 from cell 4}}$ and $\mathcal{I}_{\text{in-degree of weight 1 from cell 5}}$ changes nothing. Next, refining according to $\mathcal{I}_{\text{out-degree of weight 1 to cell 1}}$ yields

$$\mathfrak{p}_{M,8} = [\{p_3\}, \{p_2, p_4\}, \{p_1\}, \{t_{1,2}, t_{1,4}\}, \{t_{2,1}, t_{3,2}, t_{3,4}, t_{4,1}\}, \{t_{2,3}, t_{4,3}\}]$$

and refining according to $\mathcal{I}_{\text{out-degree of weight 1 to cell 2}}$ yields

$$\mathfrak{p}_{M,9} = [\{p_3\}, \{p_2, p_4\}, \{p_1\}, \{t_{1,2}, t_{1,4}\}, \{t_{2,1}, t_{4,1}\}, \{t_{3,2}, t_{3,4}\}, \{t_{2,3}, t_{4,3}\}].$$

This partition cannot be refined further by any invariant since the permutation $\begin{pmatrix} p_1 & p_2 & p_3 & p_4 & t_{1,2} & t_{2,1} & t_{2,3} & t_{3,2} & t_{3,4} & t_{4,3} & t_{4,1} & t_{1,4} \\ p_1 & p_4 & p_3 & p_2 & t_{1,4} & t_{4,1} & t_{4,3} & t_{3,4} & t_{3,2} & t_{2,3} & t_{2,1} & t_{1,2} \end{pmatrix} \in G$ is a stabilizer of M in G also fixing the partition $\mathfrak{p}_{M,9}$. ♣

6 EXPERIMENTAL RESULTS

In this section, some experimental results are given. The results are obtained by using and extending the *lola* reachability analyzer, version 1.0 beta [Schmidt 2000c].

6.1 Net Classes

The following net classes are used in the experiments.

Mutual exclusion in grid-like networks. These nets are based on the nets in [Schmidt 2000b]. A net "grid d n" models a d-dimensional hypercube of agents with n agents in each dimension. Each agent has two states, critical and non-critical, and can move from the non-critical state into critical one if none of its neighbors is in critical state. See Fig. 8 for the "grid 3 2"-net (the dotted lines are drawn only to visualize the three dimensions). The automorphism group of an d dimensional grid net is isomorphic to the automorphism group of an d-dimensional (hyper)cube.

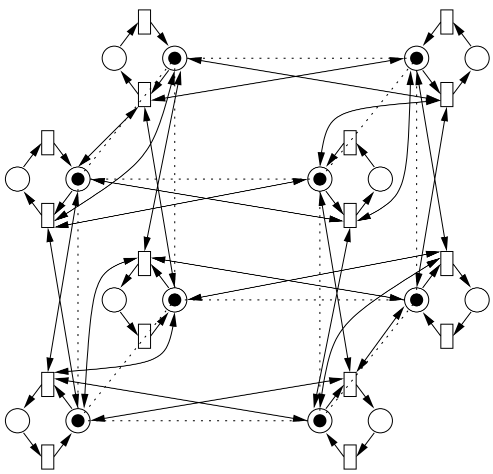


Figure 8: A three dimensional grid with two agents per row.

Dining philosophers. A version of the classic dining philosophers net. A net "ph n" has n philosophers and the automorphism group of such net is isomorphic to the cyclic group of order n.

Database managers. An unfolding of the Colored Petri net presented in [Jensen 1992]. "db n" denotes the net with n managers, having the automorphism group isomorphic to the symmetric group of degree n.

Graph enumeration nets. These nets resemble the one appearing in [Junttila 2001, Lemma 11], inspired by the system in the proof of Thm. 3.4 in [Ip 1996]. Assume a vertex set $V = \{1, \ldots, n\}$ and consider the set of all the directed, unlabeled graphs over V having no self-loops. The following net, call it "digraphs n", enumerates all such graphs in its reachable markings (see Fig. 9 for an example when $n = 3$). For each vertex $v \in V$, the net has the place p_v. Similarly, for each possible edge $\langle v_1, v_2 \rangle \in V \times V$ such that $v_1 \neq v_2$, the net has the place p_{v_1,v_2}. The purpose is that the places of form p_{v_1,v_2} describe the adjacency matrix of a graph over V and that a place p_{v_1,v_2} contains one token in a marking if and only if the graph corresponding to the marking has an edge $\langle v_1, v_2 \rangle$. For each place p_{v_1,v_2} there is a transition removing one token from it. In addition, each place p_v corresponding to a vertex v is connected to each place of form $p_{v,v'}$ with a gadget shown as a dashed line and explained in Fig. 9. Similarly, p_v is also connected to each place of form $p_{v',v}$ with a gadget shown as a dotted line and explained in Fig. 9. These gadgets guarantee that the automorphism group of the net is isomorphic to the permutation group consisting of all permutations of V (i.e., to the symmetric group of degree n). The action of a permutation π of V on the places is such that each p_v is permuted to $p_{\pi(v)}$ and each p_{v_1,v_2} is permuted to $p_{\pi(v_1),\pi(v_2)}$. Thus the action of π corresponds to the usual action of a permutation of the vertex set on the adjacency matrix of a graph. In the initial marking, all the places of form p_{v_1,v_2} corresponding to the possible edges have one token and the others are empty. Thus the set of all reachable markings of the net corresponds to the set of all directed, unlabeled graphs over V having no self-loops. Furthermore, two reachable markings are symmetric if and only if their corresponding graphs are isomorphic. Thus an optimal quotient reachability graph consisting only of one marking of each orbit has exactly one marking for each class of mutually isomorphic graphs.

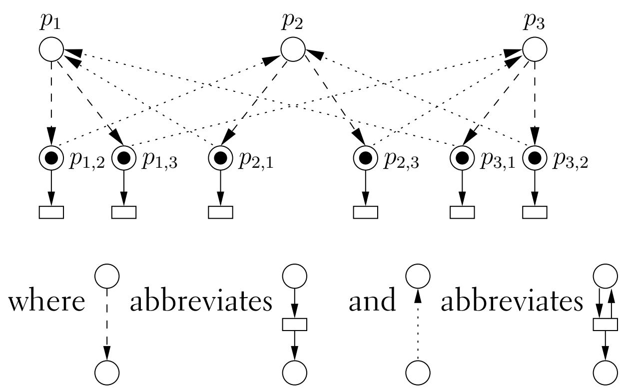


Figure 9: A net enumerating all directed graphs without self-loops over three vertices.

A similar net, call it "graphs n", enumerating all undirected, unlabeled graphs over n vertices having no self-loops can be constructed by similar principles. See Fig. 10 for an example when $n = 4$.

Properties of nets. Table 1 lists the properties of the nets used in the experiments. The columns $|P|$ and $|T|$ describe the number of places and transitions in the net, respectively, and $|G|$ gives the size of the symmetry group stabilizing the initial marking (the group that is used in the experiments). The number of reachable markings and transition firings as well as the run time of *lola* in seconds without the symmetry reduction method is given in the last three columns, respectively. For some nets the number of reachable markings is too large and running *lola* would result in running out

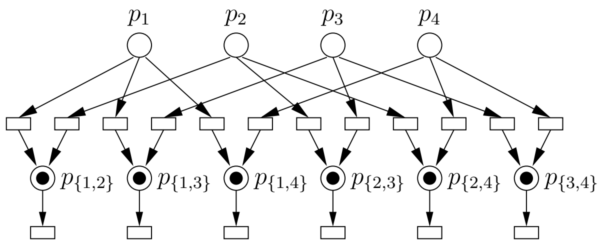


Figure 10: A net enumerating all undirected graphs without self-loops over four vertices.

net	$\|P\|$	$\|T\|$	$\|G\|$	reachable markings	edges	*lola* time
ph 10	40	30	10	6,726	43,480	1
ph 13	52	39	13	94,642	795,353	4
ph 16	64	48	16	1,331,714	13,774,112	90
db 8	193	128	40,320	17,497	81,664	1
db 9	244	162	362,880	59,050	314,946	3
db 10	301	200	3,628,800	196,831	1,181,000	15
db 20	1201	800	20!	$1+20\times(3^{20-1}) \approx 23 \times 10^9$		
grid 2 5	50	50	8	55,447	688,478	3
grid 3 3	54	54	48	70,633	897,594	4
grid 5 2	64	64	3840	254,475	3,689,792	20
graphs 5	15	30	120	1,024	5,129	1
graphs 6	21	45	720	32,678	245,760	1
graphs 7	28	63	5,040	2,097,152	22,020,096	86
graphs 8	36	84	40,320	$2^{\binom{8}{2}} = 2^{28}$		
graphs 9	45	108	362,880	$2^{\binom{9}{2}} = 2^{36}$		
digraphs 3	9	18	6	64	192	1
digraphs 4	16	36	24	4,096	24,576	1
digraphs 5	25	60	120	1,048,576	10,485,760	39
digraphs 6	36	90	720	$2^{6\times(6-1)} = 2^{30}$		

Table 1: Properties of the nets.

of memory. In such cases, the run time of *lola* is not given but the number of reachable states is given analytically.

6.2 Results

The experimental results were obtained in a PC machine with 1GHz AMD Athlon processor and 1Gb of memory, running the Debian Linux operating system. The extended *lola* was compiled with the GNU g++ compiler with the -O3 optimization flag switched on. All run-times were obtained by the Unix `time` command and are user times rounded up to full seconds unless otherwise stated. The available memory was limited to 900Mb and the available time to 24 hours by the Unix `ulimit` command.

The original symmetry reduction algorithms in *lola* described in [Schmidt 2000b] are numbered as follows: 1 refers to the "iterating the symmetries" algorithm, 2 is the "iterating the states" algorithm, and 3 is the "canonical representative" algorithm[3]. The results of these algorithms are shown in Tables 2 and 3. The *lola* implementation seems to contain some bugs since the algorithms 1 and 2 should both produce optimally reduced quotient reachability graphs but the numbers of the markings in the generated quotient reachability graphs are not the same.

	lola alg. 1			*lola* alg. 2		
net	markings	edges	time	markings	edges	time
ph 10	684	4,421	1	684	4,421	8
ph 13	7,282	61,193	2	7,282	61,193	629
ph 16	83,311	861,696	33	≥83,311	≥861,000	>24h
db 8	37	177	1	37	177	3
db 9	46	250	9	46	250	6
db 10	56	341	118	56	341	11
db 20			>24h	211	2,681	1,477
grid 2 5	7,567	94,143	1	7,471	92,982	183
grid 3 3	2,154	27,620	2	2,103	26,994	62
grid 5 2	296	4,336	7	287	4,237	14
graphs 5	34	170	1	34	170	1
graphs 6	156	1,170	1	152	1,140	1
graphs 7	1,044	10,962	17	1,022	10,731	27
graphs 8	12,346	172,844	2,358	12,095	169,330	3,662
graphs 9	>47,683	>675,000	>24h	>55,400	>792,000	>24h
digraphs 3	16	48	1	16	48	1
digraphs 4	218	1,308	1	215	1,290	1
digraphs 5	9,735	97,357	3	9,567	95,670	1,197
digraphs 6	1,598,555	24,060,959	1,810	>85,469	>908,000	>24h

Table 2: Results for the original *lola* algorithms.

Table 4 shows the results of the Schreier-Sims search algorithm described in Sect. 4. The maximal element with minimal frequency multiset selector is used because it seems to usually give the best results. For instance, the minimal element multiset selector gives for some nets bit smaller running times since it can be implemented more efficiently, but in some nets the running times are much worse. Pruning with the fixed prefix, the trivial pruning with the found stabilizers, and the base optimization described in page 18 are applied, too. The pruning of transitions with the found stabilizers was not implemented because the current *lola* implementation only stores the symmetry group restricted to the set of places. The "trivial %" and "easy %" columns show the percentage of trivial and easy canonized markings, respective, as defined in Sect. 4. The "max dead" and "av. dead" columns show the maximum and average number of dead nodes, respectively, in the search trees for hard markings. As can be seen, practically all markings are usually hard and the number of bad nodes in a search tree can grow quite large. One reason for this behavior is that all the nets are 1-safe, i.e., the number of tokens in a place in each reachable marking is at most one. Thus the multiset selector cannot usually prune the search tree efficiently.

[3]Not a canonical representative marking function by the terms used in this report.

net	*lola* alg. 3 markings	edges	time
ph 10	684	4,421	1
ph 13	7,282	61,193	1
ph 16	83,311	861,696	9
db 8	2,188	10,215	1
db 9	6,562	35,002	1
db 10	19,684	118,109	4
db 20	>399,000	>3,110,000	>418
grid 2 5	14,236	177,007	2
grid 3 3	10,847	136,446	2
grid 5 2	3,020	44,502	1
graphs 5	117	569	1
graphs 6	1,646	11,572	1
graphs 7	37,195	361,478	3
graphs 8	1,536,698	19,805,842	246
graphs 9	>5,128,600	>61,941,000	>801
digraphs 3	16	48	1
digraphs 4	347	2,038	1
digraphs 5	40,078	375,708	3
digraphs 6	>4,581,000	>56,146,000	>512

Table 3: Results for the original *lola* algorithms (continued).

Table 5 shows the results for the partition guided Schreier-Sims search algorithm described in Sect. 5. The applied partition generator first refines the unit partition according to the orbit and marking invariants and then refines the resulting partition with the partition dependent weighted in- and out-degree invariants until no improvement is achieved. For efficiency reasons, this latter refinement is implemented in a procedural way as discussed in page 24. As can be seen from the results, the amounts of trivial and easy markings are now much higher, compared to the marking guided Schreier-Sims search algorithm discussed above. Furthermore, the hard markings are also easier, and although the number of dead nodes can be still in thousands, on the average it is very low. For nets with small symmetry groups, the overhead of computing the ordered partition sometimes makes the algorithm slower than the marking guided Schreier-Sims search (e.g., the dining philosophers nets and the nets “grid 2 5”, “grid 3 3”, and “digraphs 6”).

Table 6 shows the results of the characteristic graph approach described in Sect. 3 when *nauty* [McKay 1990] is used as the graph canonizer. The “trivial %” column shows the percentage of the trivial canonized markings, i.e., markings for which the search tree of *nauty* contains only one node. The “max nodes” and “av. nodes” columns give the maximum and average number of *nauty* search tree nodes, respectively, for the canonized non-trivial markings. Note that the percentage of the trivial markings encountered is essentially the same as in the partition guided Schreier-Sims search approach discussed above. This not a surprise since the preprocessing technique in *nauty* and the applied partition generator are based on the same ideas (recall Sect. 5). Note that although the search tree sizes of *nauty* are very small in all

net	markings	edges	time	trivial %	easy %	max dead	av. dead
ph 10	684	4,421	1	7.71	0.50	9	1.38
ph 13	7,282	61,193	1	2.85	0.00	12	2.09
ph 16	83,311	861,696	15	1.04	0.01	15	2.98
db 8	37	177	1	0	6.21	66	24.98
db 9	46	250	1	0	4.80	132	39.49
db 10	56	341	1	0	3.81	259	60.36
db 20	211	2,681	172	0	0.86	40,152	1,844.12
grid 2 5	7,471	92,982	1	0	3.88	7	2.07
grid 3 3	2,103	26,994	1	0	1.59	46	12.14
grid 5 2	288	4,253	1	0	1.03	278	126.94
graphs 5	34	170	1	0	0.59	61	38.25
graphs 6	156	1,170	1	0	0.09	313	109.76
graphs 7	1,044	10,962	3	0	0.01	1,413	272.48
graphs 8	12,346	172,844	82	0	0.00	8,770	580.28
graphs 9	274,668	4,944,024	5,036	0	0.00	70,017	226.80
digraphs 3	16	48	1	29.17	22.92	2	1.17
digraphs 4	218	1,308	1	0	4.05	7	3.15
digraphs 5	9,608	96,080	2	0	0.09	27	7.97
digraphs 6	1,540,944	23,114,160	929	0	0.00	93	17.19

Table 4: Results of the plain Schreier-Sims search.

examples, the running times are high. The bad running times are because of the following:

1. *nauty* does not handle edge labels and is optimized for undirected graphs. P/T-nets are, on the other hand, edge labeled and directed. Thus some extra vertices have to be included in the graphs (recall Sect. 3).
2. While P/T-nets are usually sparse, *nauty* is designed for dense graphs in a sense that the graphs are internally represented as adjacency matrixes. Thus storing a graph with thousands of vertices takes a lot of memory and consequently slows down the refinement routines needed during the reach tree traversal in *nauty*.

The results would probably look very different if a graph canonizer designed for (i) sparse, and (ii) vertex and edge labeled directed graphs were used.

7 CONCLUSIONS

In this report, three new algorithm for producing (canonical) representative markings for Place/Transition-nets are described. All the algorithms require the precomputation of a Schreier-Sims representation of the symmetry group of the net in question. The first algorithm builds a characteristic graph of the marking and obtains the canonical representative for the marking from the canonical version of the graph, which is produced by a black box graph canonizer. The second algorithm is a backtrack search in the Schreier-Sims rep-

net	markings	edges	time	trivial %	easy %	max dead	av. dead
ph 10	684	4,421	1	98.76	1.24	-	-
ph 13	7,282	61,193	4	99.997	0.003	-	-
ph 16	83,311	861,696	66	99.91	0.09	-	-
db 8	37	177	1	0	100	-	-
db 9	46	250	1	0	100	-	-
db 10	56	341	1	0	100	-	-
db 20	211	2,681	27	0	100	-	-
grid 2 5	7,471	92,982	9	90.86	8.71	2	1.19
grid 3 3	2,103	26,994	4	60.82	33.22	16	1.86
grid 5 2	288	4,253	1	2.26	72.00	15	2.30
graphs 5	34	170	1	0	71.18	16	2.29
graphs 6	156	1,170	1	11.11	57.78	40	3.31
graphs 7	1,044	10,962	1	24.70	50.58	196	3.18
graphs 8	12,346	172,844	15	40.52	42.95	535	3.71
graphs 9	274,668	4,944,024	586	57.46	33.92	2,045	3.02
digraphs 3	16	48	1	77.08	22.92	-	-
digraphs 4	218	1,308	1	78.29	21.18	3	2.29
digraphs 5	9,608	96,080	7	89.15	10.22	10	1.52
digraphs 6	1,540,944	23,114,160	2,404	95.68	4.05	34	1.10

Table 5: Results of the partition guided Schreier-Sims search.

resentation of the symmetry group. The search tree is pruned by the marking in question and the stabilizers of the marking (which are found during the search). The algorithm returns the smallest marking it finds as the canonical representative. The third algorithm improves the second one by combining it with a standard preprocessing technique applied in graph isomorphism/canonizer algorithms. That is, an ordered partition for the marking is first built by applying a set of invariants and the partition is then used to further prune the search in the Schreier-Sims representation. The second and third algorithm could be approximated (i.e., made non-canonical) by performing only a limited search in the Schreier-Sims representation. For instance, an upper limit for the traversed nodes could be set.

Some experimental results are provided, too. They show that the proposed algorithms are competitive against the existing algorithms described in [Schmidt 2000a; 2000b]. The third proposed algorithm is perhaps the most robust one, working well with many kinds of symmetry groups, even with very large ones. However, an efficient graph canonizer designed especially for the class of sparse, vertex and edge labeled directed graphs would probably make the first algorithm competitive, too.

An interesting alternative for producing canonical representative markings not discussed earlier in this report is the string canonization algorithm in [Babai and Luks 1983]. The algorithm does the canonization orbit-wise, and also exploits the imprimitivity of groups. However, the algorithm seems to involve more complex permutation group algorithms and implementing it is left as a future challenge.

net	markings	edges	time	trivial %	max nodes	av. nodes
ph 10	684	4,421	8	98.76	3	3.00
ph 13	7,282	61,193	201	99.997	3	3.00
ph 16	83,311	861,696	4,866	99.91	3	3.00
db 8	37	177	87	0	36	18.09
db 9	46	250	278	0	45	24.11
db 10	56	341	877	0	55	31.12
db 20			>24h			
grid 2 5	7,471	92,982	2,303	90.86	8	3.13
grid 3 3	2,103	26,994	1,498	60.82	10	3.52
grid 5 2	288	4,253	920	2.26	21	5.71
graphs 5	34	170	1	0	15	5.02
graphs 6	156	1,170	3	11.11	21	5.47
graphs 7	1,044	10,962	38	24.70	28	5.29
graphs 8	12,346	172,844	1,053	40.52	36	4.83
graphs 9	274,668	4,944,024	49,916	57.46	45	4.26
digraphs 3	16	48	1	77.08	4	3.09
digraphs 4	218	1,308	2	78.29	8	3.23
digraphs 5	9,608	96,080	243	89.15	13	3.23
digraphs 6	>1,028,419	>14,187,000	>24h			

Table 6: Results of the characteristic graph approach.

Acknowledgements

The author wishes to thank Petteri Kaski for his comments on this work. The financial support of the Helsinki Graduate School in Computer Science and Engineering (HeCSE) and the Academy of Finland (projects no. 47754 and no. 53695) is also 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, Boston, Massachusetts, April 25–27*. ACM, 171–183.

BUTLER, G. 1991. *Fundamental Algorithms for Permutation Groups*. Lecture Notes in Computer Science, vol. 559. Springer.

CHIOLA, G., DUTHEILLET, C., FRANCESCHINIS, G., AND HADDAD, S. 1991. On well-formed coloured nets and their symbolic reachability graph. See Jensen and Rozenberg [1991], 373–396.

CLARKE, E. M., EMERSON, E. A., JHA, S., AND SISTLA, A. P. 1998. Symmetry reductions in model checking. In *Computer Aided Verification: 10th International Conference, CAV'98*, A. J. Hu and M. Y. Vardi, Eds. Lecture Notes in Computer Science, vol. 1427. Springer, 147–158.

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.

EMERSON, E. A. AND SISTLA, A. P. 1997. Utilizing symmetry when model checking under fairness assumptions: An automata-theoretic approach. *ACM Transactions on Programming Languages and Systems 19,* 4 (July), 617–638.

GENRICH, H. J. 1991. Predicate/transition nets. See Jensen and Rozenberg [1991], 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. Tech. Rep. DAIMI PB 174, Datalogisk Afdeling, Matematisk Institut, Aarhus Universitet. May.

IP, C. N. 1996. State reduction methods for automatic formal verification. Ph.D. thesis, Department of Computer Science, Stanford University.

IP, C. N. AND DILL, D. L. 1996. Better verification through symmetry. *Formal Methods in System Design 9,* 1/2 (Aug.), 41–76.

JENSEN, K. 1992. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use: Volume 1, Basic Concepts*, Second ed. Monographs in Theoretical Computer Science. Springer.

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

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

JENSEN, K. AND ROZENBERG, G., Eds. 1991. *High-level Petri Nets; Theory and Application.* Springer.

JERRUM, M. 1986. A compact representation for permutation groups. *Journal of Algorithms 7,* 1 (Mar.), 60–78.

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.

JUNTTILA, T. 2002. Symmetry reduction algorithms for data symmetries. Research Report A72, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland. May.

JUNTTILA, T. A. 2001. Computational complexity of the Place/Transition-net symmetry reduction method. *Journal of Universal Computer Science 7,* 4, 307–326.

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

LORENTSEN, L. AND KRISTENSEN, L. M. 2001. Exploiting stabilizers and parallelism in state space generation with the symmetry method. In *Proceedings of the Second International Conference on Application of Concurrency to System Design (ACSD 2001).* IEEE Computer Society, 211–220.

MCKAY, B. D. 1981. Practical graph isomorphism. *Congressus Numerantium 30,* 45–87.

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

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. 2000c. LoLA: A low level analyser. In *Application and Theory of Petri Nets 2000; Proceedings of the 21st International Conference, ICATPN 2000; Aarhus, Denmark, June 2000,* M. Nielsen and D. Simpson, Eds. Lecture Notes in Computer Science, vol. 1825. Springer, 465–474.

SISTLA, A. P., GYURIS, V., AND EMERSON, E. A. 2000. SMC: A symmetry-based model checker for verification of safety and liveness properties. *ACM Transactions on Software Engineering and Methodology* 9, 2 (Apr.), 133–166.

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

A PROOFS

Proof of Theorem 2.5

Theorem 2.5 *Let $g \in G$. A permutation $\hat{g} \in G$ is compatible with a place valuation pval if and only if the permutation $g \circ \hat{g} \in G$ is compatible with the permuted place valuation $g(pval)$.*

Proof. Assume that $\hat{g}_1 \circ \cdots \circ \hat{g}_{|P|} \circ \hat{g}_{|P|+1} \cdots \circ \hat{g}_{|P|+|T|}$ is the unique representation of $\hat{g}$ in the fixed Schreier-Sims representation of G. Similarly, let $\hat{g}'_1 \circ \cdots \hat{g}'_{|P|} \circ \hat{g}'_{|P|+1} \cdots \circ \hat{g}'_{|P|+|T|}$ be the unique representation of $\hat{g}' = g \circ \hat{g}$. Fix any i, $1 \leq i \leq |P|$. It must now be proven that

$$pval((\hat{g}_1 \circ \cdots \circ \hat{g}_i)(p_{\beta,i})) \in select\left(\sum_{h \in U_i} 1'pval(\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\})\right)$$

if and only if

$$(g(pval))((\hat{g}'_1 \circ \cdots \circ \hat{g}'_i)(p_{\beta,i})) \in select\left(\sum_{h \in U_i} 1'(g(pval))(\{(\hat{g}'_1 \circ \cdots \circ \hat{g}'_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\})\right).$$

First, $(\hat{g}_1 \circ \cdots \circ \hat{g}_i)(p_{\beta,i}) = \hat{g}(p_{\beta,i})$ because the "postfix" permutation $\hat{g}_{i+1} \circ \cdots \circ \hat{g}_{|P|}$ of $\hat{g}$ fixes $p_{\beta,i}$. Similarly, $(\hat{g}'_1 \circ \cdots \circ \hat{g}'_i)(p_{\beta,i}) = \hat{g}'(p_{\beta,i}) = (g \circ \hat{g})(p_{\beta,i}) = g(\hat{g}(p_{\beta,i}))$. Thus $pval((\hat{g}_1 \circ \cdots \circ \hat{g}_i)(p_{\beta,i})) = pval(\hat{g}(p_{\beta,i}))$, $(g(pval))((\hat{g}'_1 \circ \cdots \circ \hat{g}'_i)(p_{\beta,i})) = (g(pval))(\hat{g}'(p_{\beta,i})) = (g(pval))(g(\hat{g}(p_{\beta,i}))) = pval(\hat{g}(p_{\beta,i}))$ and

$$pval((\hat{g}_1 \circ \cdots \circ \hat{g}_i)(p_{\beta,i})) = (g(pval))((\hat{g}'_1 \circ \cdots \circ \hat{g}'_i)(p_{\beta,i})).$$

Secondly, note that $\{h(p_{\beta,i}) \mid h \in U_i\} = [p_{\beta,i}]_{G_{i-1}}$, i.e. the orbit of $p_{\beta,i}$ under G_{i-1}, and thus $\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\}$ equals to $(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1})([p_{\beta,i}]_{G_{i-1}})$. As the last permutations $\hat{g}_i \circ \cdots \circ \hat{g}_{|P|}$ in the representation of $\hat{g}$ belong to the subgroup G_{i-1}, $(\hat{g}_i \circ \cdots \circ \hat{g}_{|P|})([p_{\beta,i}]_{G_{i-1}}) = [p_{\beta,i}]_{G_{i-1}}$. Thus $\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\} = (\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1})([p_{\beta,i}]_{G_{i-1}}) = \hat{g}([p_{\beta,i}]_{G_{i-1}})$. Similarly, $\{(\hat{g}'_1 \circ \cdots \circ \hat{g}'_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\} = \hat{g}'([p_{\beta,i}]_{G_{i-1}}) = (g \circ \hat{g})([p_{\beta,i}]_{G_{i-1}}) = g\left(\hat{g}([p_{\beta,i}]_{G_{i-1}})\right)$ Therefore,

$$\{(\hat{g}'_1 \circ \cdots \circ \hat{g}'_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\} = g\left(\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\}\right).$$

This implies that

$$\sum_{h \in U_i} 1'(g(pval))(\{(\hat{g}'_1 \circ \cdots \circ \hat{g}'_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\}) =$$
$$\sum_{h \in U_i} 1'(g(pval))(g\left(\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\}\right)) =$$
$$\sum_{h \in U_i} 1'pval(\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\})$$

and thus

$$select(\sum_{h \in U_i} 1'(g(pval))(\{(\hat{g}'_1 \circ \cdots \circ \hat{g}'_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\})) =$$
$$select(\sum_{h \in U_i} 1'pval(\{(\hat{g}_1 \circ \cdots \circ \hat{g}_{i-1} \circ h)(p_{\beta,i}) \mid h \in U_i\})).$$

□

Proof of Theorem 3.1

Theorem 3.1 *The mapping $\mathcal{K}_{\mathbb{M}}$ is a canonical representative function.*

Proof. Clearly $\mathcal{K}_{\mathbb{M}}(M)$ is symmetric to M under G because $\mathcal{K}_{\mathbb{M}}(M)$ is obtained by applying an element in G to M.

Assume two markings, M_1 and M_2, that are symmetric under G. By definition, their characteristic graphs, $\mathcal{G}_{M_1}$ and $\mathcal{G}_{M_2}$, respectively, are isomorphic. Assume that $\mathcal{K}(\mathcal{G}_{M_1})$ (which equals to $\mathcal{K}(\mathcal{G}_{M_2})$) is the canonical version of $\mathcal{G}_{M_1}$ and $\mathcal{G}_{M_2}$. Take any canonization mapping (i.e. isomorphism) γ_1 from $\mathcal{G}_{M_1}$ to $\mathcal{K}(\mathcal{G}_{M_1})$ and γ_2 from $\mathcal{G}_{M_2}$ to $\mathcal{K}(\mathcal{G}_{M_1})$. Now $\gamma_2^{-1} \circ \gamma_1$ is an isomorphism from $\mathcal{G}_{M_1}$ to $\mathcal{G}_{M_2}$ and $\gamma_1^{-1} \circ \gamma_2$ is an isomorphism from $\mathcal{G}_{M_2}$ to $\mathcal{G}_{M_1}$. By the definition of characteristic graphs, $\gamma_2^{-1} \circ \gamma_1$ restricted to $P \cup T$ belongs to G and maps M_1 to M_2 and $\gamma_1^{-1} \circ \gamma_2$ restricted to $P \cup T$ belongs to G and maps M_2 to M_1.

Define the place valuations $pval_1$ and $pval_2$ by $\forall p \in P : pval_1(p) = \gamma_1(p)$ and $\forall p \in P : pval_2(p) = \gamma_2(p)$. Now $\left((\gamma_2^{-1} \circ \gamma_1)(pval_1)\right)(p) = pval_1((\gamma_2^{-1} \circ \gamma_1)^{-1}(p)) = pval_1(\gamma_1^{-1}(\gamma_2(p))) = \gamma_1(\gamma_1^{-1}(\gamma_2(p))) = \gamma_2(p) = pval_2(p)$ i.e. $\gamma_2^{-1} \circ \gamma_1$ restricted to $P \cup T$ maps $pval_1$ to $pval_2$.

Observe that $pval_1$ and $pval_2$ are clearly injective functions. Assume that $\hat{g}_1$ is the unique element in G that is compatible with $pval_1$. By Thm. 2.5, $\hat{g}_1$ is compatible with $pval_1$ if and only if $(\gamma_2^{-1} \circ \gamma_1) \circ \hat{g}_1$ is compatible with $pval_2$. Thus $(\gamma_2^{-1} \circ \gamma_1) \circ \hat{g}_1$ is the unique element in G that is compatible with $pval_2$. Now $((\gamma_2^{-1} \circ \gamma_1) \circ \hat{g}_1)^{-1}(M_2) = \hat{g}_1^{-1}((\gamma_1^{-1} \circ \gamma_2)(M_2)) = \hat{g}_1^{-1}(M_1)$ and thus $\mathcal{K}_{\mathbb{M}}(M_1) = \mathcal{K}_{\mathbb{M}}(M_2)$.

The fact that $\mathcal{K}_{\mathbb{M}}(M)$ is uniquely determined follows by considering the case $M_1 = M_2$. □

Proof of Theorem 4.1

Theorem 4.1 *For each marking $M \in \mathbb{M}$ and for each symmetry $g \in G$, $posreps(M) = posreps(g(M))$.*

Proof. By Thm. 2.5, $\hat{g}$ is compatible with M if and only if $g \circ \hat{g}$ is compatible with $g(M)$. In addition, $(g \circ \hat{g})^{-1}(g(M)) = \hat{g}^{-1}(g^{-1}(g(M))) = \hat{g}^{-1}(M)$. □

Proof of Lemma 5.2

Lemma 5.2 *For all $g \in G$ and all markings M, $pval_{f(g(M))} = g(pval_{f(M)})$.*

Proof. For each place $p \in P$,

$$\begin{aligned} pval_{f(g(M))}(p) &= incell(f(g(M)), p) \\ &= incell(g(f(M)), p) \\ &= incell(f(M), g^{-1}(p)) \\ &= \left(pval_{f(M)}\right)(g^{-1}(p)) \\ &= \left(g(pval_{f(M)})\right)(p). \end{aligned}$$

□

Proof of Lemma 5.4

Lemma 5.4 *If $incell(f(M), p_1) = incell(f(M), p_2) \Rightarrow M(p_1) = M(p_2)$ for all places $p_1, p_2 \in P$, for all places $p_1, p_2 \in P$, then $\mathrm{Stab}(G, pval_{f(M)}) = \mathrm{Stab}(G, M)$.*

Proof. In Cor. 5.3, it is shown that $\mathrm{Stab}(G, M) \subseteq \mathrm{Stab}(G, pval_{f(M)})$.

Take any permutation $g \in \mathrm{Stab}(G, pval_{f(M)})$, any place $p \in P$ and assume that $incell(f(M), p_1) = incell(f(M), p_2)$ implies $M(p_1) = M(p_2)$ for all places $p_1, p_2 \in P$. It is now shown that

$$(g(M))(p) = M(p)$$

which is enough to imply that $g(M) = M$ i.e. $g \in \mathrm{Stab}(G, M)$ and thus that $\mathrm{Stab}(G, pval_{f(M)}) \subseteq \mathrm{Stab}(G, M)$. Since g is a stabilizer of $pval_{f(M)}$ in G, $g(pval_{f(M)}) = pval_{f(M)}$ holds and implies that

$$(g(pval_{f(M)}))(p) = pval_{f(M)}(p). \tag{1}$$

By the action of g on $pval_{f(M)}$, $(g(pval_{f(M)}))(p) = pval_{f(M)}(g^{-1}(p))$, which combined with (1) gives $pval_{f(M)}(p) = pval_{f(M)}(g^{-1}(p))$. Applying the definition of $pval_{f(M)}$ to this gives $incell(f(M), p) = incell(f(M), g^{-1}(p))$. Thus by the initial assumption it holds that $M(p) = M(g^{-1}(p))$ implying that $(g(M))(p) = M(g^{-1}(p)) = M(p)$, which concludes the proof. □

Proof of Theorem 5.5

Theorem 5.5 *For each marking $M \in \mathbb{M}$ and for each symmetry $g \in G$, $posreps(M) = posreps(g(M))$.*

Proof. By Thm 2.5 and Lemma 5.2, $\hat{g}$ is compatible with $pval_f(M)$ if and only if $g \circ \hat{g}$ is compatible with $g(pval_{f(M)}) = pval_{f(g(M))}$. In addition, $(g \circ \hat{g})^{-1}(g(M)) = \hat{g}^{-1}(g^{-1}(g(M))) = \hat{g}^{-1}(M)$. □

Proof of Lemma 5.9

Lemma 5.9 *For each G-partition refiner $\mathcal{R}$, the function $f_{\mathcal{R}} : \mathbb{M} \rightarrow \mathfrak{P}$ defined by $f_{\mathcal{R}}(M) = \mathcal{R}(M, [P \cup T])$ is a G-partition generator.*

Proof. For each $g \in G$, $f_{\mathcal{R}}(g(M)) = \mathcal{R}(g(M), [P \cup T]) = \mathcal{R}(g(M), g([P \cup T])) = g(\mathcal{R}(M, [P \cup T]) = g(f_{\mathcal{R}}(M))$. □

Proof of Lemma 5.11

Lemma 5.11 *The function $\mathcal{R}_{\mathcal{I}}$ is a G-partition refiner.*

Proof. The fact that $\mathcal{R}_{\mathcal{I}}(M, \mathfrak{p}) \preceq \mathfrak{p}$ is straightforward to see. Take any $g \in G$, any marking M, and any partition $\mathfrak{p}$. Assume that $\mathcal{R}_{\mathcal{I}}(M, \mathfrak{p}) = \mathfrak{p}_{r,1}$ and $\mathcal{R}_{\mathcal{I}}(g(M), g(\mathfrak{p})) = \mathfrak{p}_{r,2}$. It remains to be shown that $g(\mathfrak{p}_{r,1}) = \mathfrak{p}_{r,2}$. For all $x, x' \in P \cup T$,

$$\begin{aligned}
& \mathit{incell}(g(\mathfrak{p}_{r,1}), x) = \mathit{incell}(g(\mathfrak{p}_{r,1}), x') \\
\Leftrightarrow\ & \mathit{incell}(\mathfrak{p}_{r,1}, g^{-1}(x)) = \mathit{incell}(\mathfrak{p}_{r,1}, g^{-1}(x')) \\
\Leftrightarrow\ & \mathit{incell}(\mathfrak{p}, g^{-1}(x)) = \mathit{incell}(\mathfrak{p}, g^{-1}(x')) \text{ and} \\
& \mathcal{I}(M, \mathfrak{p}, g^{-1}(x)) = \mathcal{I}(M, \mathfrak{p}, g^{-1}(x')) \\
\Leftrightarrow\ & \mathit{incell}(g(\mathfrak{p}), x) = \mathit{incell}(g(\mathfrak{p}), x') \text{ and} \\
& \mathcal{I}(g(M), g(\mathfrak{p}), x) = \mathcal{I}(g(M), g(\mathfrak{p}), x') \\
\Leftrightarrow\ & \mathit{incell}(\mathfrak{p}_{r,2}, x) = \mathit{incell}(\mathfrak{p}_{r,2}, x')
\end{aligned}$$

and thus the cells in $g(\mathfrak{p}_{r,1})$ and in $\mathfrak{p}_{r,2}$ are the same. Similarly, for all $x, x' \in P \cup T$,

$$\begin{aligned}
& \mathit{incell}(g(\mathfrak{p}_{r,1}), x) < \mathit{incell}(g(\mathfrak{p}_{r,1}), x') \\
\Leftrightarrow\ & \mathit{incell}(\mathfrak{p}_{r,1}, g^{-1}(x)) < \mathit{incell}(\mathfrak{p}_{r,1}, g^{-1}(x')) \\
\Leftrightarrow\ & (a) \quad \mathit{incell}(\mathfrak{p}, g^{-1}(x)) < \mathit{incell}(\mathfrak{p}, g^{-1}(x')) \text{ or} \\
& (b) \quad \mathit{incell}(\mathfrak{p}, g^{-1}(x)) = \mathit{incell}(\mathfrak{p}, g^{-1}(x')) \text{ and} \\
& \qquad \mathcal{I}(M, \mathfrak{p}, g^{-1}(x)) < \mathcal{I}(M, \mathfrak{p}, g^{-1}(x')) \\
\Leftrightarrow\ & (a) \quad \mathit{incell}(g(\mathfrak{p}), x) < \mathit{incell}(g(\mathfrak{p}), x') \text{ or} \\
& (b) \quad \mathit{incell}(g(\mathfrak{p}), x) = \mathit{incell}(g(\mathfrak{p}), x') \text{ and} \\
& \qquad \mathcal{I}(g(M), g(\mathfrak{p}), x) < \mathcal{I}(g(M), g(\mathfrak{p}), x') \\
\Leftrightarrow\ & \mathit{incell}(\mathfrak{p}_{r,2}, x) < \mathit{incell}(\mathfrak{p}_{r,2}, x')
\end{aligned}$$

and thus the cells in $g(\mathfrak{p}_{r,1})$ and in $\mathfrak{p}_{r,2}$ are ordered in the same way. Therefore, $g(\mathfrak{p}_{r,1}) = \mathfrak{p}_{r,2}$. □

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A62 Kari J. Nurmela, Patric R. J. Östergård
Covering a Square with up to 30 Equal Circles. June 2000.

HUT-TCS-A63 Nisse Husberg, Tomi Janhunen, Ilkka Niemelä (Eds.)
Leksa Notes in Computer Science. October 2000.

HUT-TCS-A64 Tuomas Aura
Authorization and availability - aspects of open network security. November 2000.

HUT-TCS-A65 Harri Haanpää
Computational Methods for Ramsey Numbers. November 2000.

HUT-TCS-A66 Heikki Tauriainen
Automated Testing of Büchi Automata Translators for Linear Temporal Logic. December 2000.

HUT-TCS-A67 Timo Latvala
Model Checking Linear Temporal Logic Properties of Petri Nets with Fairness Constraints. January 2001.

HUT-TCS-A68 Javier Esparza, Keijo Heljanko
Implementing LTL Model Checking with Net Unfoldings. March 2001.

HUT-TCS-A69 Marko Mäkelä
A Reachability Analyser for Algebraic System Nets. June 2001.

HUT-TCS-A70 Petteri Kaski
Isomorph-Free Exhaustive Generation of Combinatorial Designs. December 2001.

HUT-TCS-A71 Keijo Heljanko
Combining Symbolic and Partial Order Methods for Model Checking 1-Safe Petri Nets. February 2002.

HUT-TCS-A72 Tommi Junttila
Symmetry Reduction Algorithms for Data Symmetries. May 2002.

HUT-TCS-A73 Toni Jussila
Bounded Model Checking for Verifying Concurrent Programs. August 2002.

HUT-TCS-A74 Sam Sandqvist
Aspects of Modelling and Simulation of Genetic Algorithms: A Formal Approach. September 2002.

HUT-TCS-A75 Tommi Junttila
New Canonical Representative Marking Algorithms for Place/Transition-Nets. October 2002.

ISBN 951-22-6218-5
ISSN 1457-7615