On Computational Complexity of Automorphism Groups in Classical Planning

Alexander Shleyfman
Technion, Haifa, Israel
alesh@campus.technion.ac.il

Abstract

Symmetry-based pruning is a family of state-of-the art methods that are used to reduce the search effort. Applying these methods requires first to establish an automorphism group that is used later on during the main search procedure. Although this group can be applied in various contexts, one of the prominent ways is to use it for pruning symmetric states. Despite the increasing popularity of these techniques, nothing have been said about the computational complexity of the automorphism group of a general planning task. Herein, we show a reduction that proves that the aforementioned problem of computing the symmetry group of planning task is GI-hard. Furthermore, we discuss the presentation of these symmetry groups and list some of their drawbacks.

Introduction

Symmetry breaking is a method for search reduction, that was well-explored across several areas in Computer Science, including, but not limited to classical planning (e. g. (Starke 1991; Emerson and Sistla 1996; Fox and Long 1999; Rintanen 2003; Pochter *et al.* 2011; Domshlak *et al.* 2012)). Symmetry pruning divides the states in the search space into orbit-based equivalence classes, which in turn, allows to explore only one representative state per such class. Application of this technique to the forward search partially prunes the exponential growth of the search space in the presence of objects with symmetric behavior. This method, however, requires to compute first some subgroup of automorphisms of the state transition graph.

The first notion of symmetries for classical planing was proposed by Pochter *et al.* (2011), and then refined by Domshlak *et al.* (2012). The definitions presented in these works, practical however they are, were based on the notion of colored graphs, and thus are quite cumbersome to reason about. Later on, Shleyfman *et al.* (2015) came up with the notion of structural symmetries that captures previously proposed concepts, and which can be derived from the syntax of a planning task in a simple declarative manner.

With this in mind, it is quite surprising, that not much has been said about the complexity of computing an automorphism group for a given planning task. In this work, we present two reductions that not only provide us with a pessimistic result that computing automorphism group for a specific planning task is as hard as for any undirected graph, but also show the non trivial connection between the automorphism groups of a planning task and its causal graph.

Background

To define a *planning task* we use the finite-domain representation formalism (FDR) presented by (Bäckström and Nebel 1995; Helmert 2006). Each task is given by a tuple $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$, where $\mathcal{V}$ is a set of multivalued *variables*, each associated with a finite domain $\mathcal{D}(v)$. The sets of variable/value pairs are written as $\langle var, val \rangle$, and sometimes referred as *facts*. A *state* s is a full variable assignments which maps each variable $v \in \mathcal{V}$ to some value in its domain, i.e. $s(v) \in \mathcal{D}(v)$. For $V \subseteq \mathcal{V}$, $s[V]$ denotes the partial assignment (also referred as a *partial state*) of s over V. *Initial state* I is a state. The *goal* G is a partial assignment. Let p be a partial assignment. We denote by $vars(p) \subseteq \mathcal{V}$ the subset of variables on which p is defined. For two partial assignments p and q, we say that p *satisfies* q, if $vars(q) \subseteq vars(p)$, and $p[v] = q[v]$ for all $v \in vars(q)$, this is denoted by $p \models q$. $\mathcal{A}$ is a finite set of *actions*, each represented by a triple $\langle \mathsf{pre}(a), \mathsf{eff}(a), \mathsf{cost}(a) \rangle$ of *precondition*, *effect*, and *cost*, where $\mathsf{pre}(a)$ and $\mathsf{eff}(a)$ are partial assignments to $\mathcal{V}$, and $\mathsf{cost}(a) \in \mathbb{R}^{0+}$. In this work we assume all actions are of unit-cost, unless otherwise stated. An action a is *applicable* in a state s if $s \models \mathsf{pre}(a)$. Applying a in s changes the value of all $v \in vars(\mathsf{eff}(a))$ to $\mathsf{eff}(a)[v]$, and leaves s unchanged elsewhere. The outcome state is denoted by $s[\![a]\!]$.

S denotes the set of all states of Π. We say that action sequence π is a *plan*, if it begins in I ends in in s_G s.t. $s_G \models G$, and each action in π is iteratively applicable, i.e. for each $a_i \in \pi$ holds that $s_{i-1} \models \mathsf{pre}(a_i)$ and $s_{i-1}[\![a_i]\!] = s_i$. The cost of a plan defined as $\mathsf{cost}(\pi) = \sum_{a_i \in \pi} \mathsf{cost}(a_i)$. An *optimal* plan, is a plan of a minimal cost. In a unit-cost domain, an optimal plan is a plan of the shortest length. The *state space* of Π is denoted $\mathcal{T}_\Pi$.

A *directed graph* is pair $\langle N, E \rangle$ where N is the set of *vertices*, and $E \subseteq N^2$ is a set of *edges*. An *undirected graph* is a pair $\langle N, E \rangle$ where N, once again, is the set of *vertices*, and $E \subseteq \{e \subseteq N \mid |e| = 2\}$ is the set of edges.

Let $\mathcal{G} = \langle N, E \rangle$ be a (un-)directed graph, and let σ be a permutation over the vertices N, we say that σ is a graph *automorphism* (or just an automorphism, if this is clear from the context) when $(n, n') \in E$ iff $(\sigma(n), \sigma(n')) \in E$. The automorphisms[1] of a graph $\mathcal{G}$ are closed under composition, and for every automorphism there exists an inverse permutation which is also an automorphism. Thus, automorphisms of a graph form a group. We call this group an *automorphism group*, and denote it by $Aut(\mathcal{G})$. The identity element e in this group will be denoted by $id_{\mathcal{G}}$.

Causal Graph

One way of capturing the structural complexity of a planning task are causal graphs. The idea mention in numerous papers, e.g. (Knoblock 1994; Bacchus and Yang 1994; Domshlak and Brafman 2002), but here we will follow the definition given by Helmert (2010).

The *causal graph* of a planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$ is a directed graph $CG(\Pi) = \langle \mathcal{V}, \mathcal{E} \rangle$, where $(u, v) \in \mathcal{E}$ if $u \neq v$ and there exists $a \in \mathcal{A}$, s.t. $u \in vars(\mathsf{pre}(a)) \cup vars(\mathsf{eff}(a))$ and $v \in vars(\mathsf{eff}(a))$.

In a nutshell, the causal graph contains an edge from a source variable to a target variable if changing the value of the target variable may depend on the value of the source variable, even if it's only a co-depending effects.

Structural Symmetries

The second ingredient we need was recently introduced by Shleyfman *et al.* (2015). This subsection defines the notion of *structural symmetries*, which capture previously proposed concepts of symmetries in classical planning. In short, structural symmetries are relabellings of the FDR representation of a given planning task Π. Variables are mapped to variables, values to values (preserving the $\langle var, val \rangle$ structure), and actions are mapped to actions. In this work, we follow the definition of structural symmetries for FDR planning tasks as defined by Wehrle *et al.* (2015). For a planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$, let P be the set of Π's facts, and let $P_{\mathcal{V}} := \{\{\langle v, d \rangle \mid d \in \mathcal{D}(v)\} \mid v \in \mathcal{V}\}$ be the set of sets of facts attributed to each variable in $\mathcal{V}$. We say that a permutation $\sigma : P \cup A \to P \cup \mathcal{A}$ is a *structural symmetry* if the following holds:

1. $\sigma(P_{\mathcal{V}}) = P_{\mathcal{V}}$,
2. $\sigma(\mathcal{A}) = \mathcal{A}$, and, for all $a \in \mathcal{A}$, $\sigma(\mathsf{pre}(a)) = \mathsf{pre}(\sigma(a))$, $\sigma(\mathsf{eff}(a)) = \mathsf{eff}(\sigma(a))$, and $\mathsf{cost}(\sigma(a)) = \mathsf{cost}(a)$.
3. $\sigma(G) = G$.

Note, that that the definition $\sigma(X) := \{\sigma(x) \mid x \in X\}$, where X is a set, can be applied recursively. For example, let s be a partial state, since s can be represented a set of facts, applying σ to s will result in a partial state s', s.t. for all facts $\langle v, d \rangle \in s$ it holds that $\sigma(\langle v, d \rangle) = \langle v', d' \rangle \in s'$ and $s'[v'] = d'$.

A set of structural symmetries Σ for a planning task Π induce a subgroup Γ of the automorphism group $Aut(\mathcal{T}_{\Pi})$, which in turn defines an equivalence relation over the states S of Π. Namely, we say that s is *symmetric* to s' iff there exists an automorphism $\sigma \in \Gamma$ such that $\sigma(s) = s'$.

[1]The definition of a *graph automorphism* for an undirected graph is almost the same with the only exception of edges, that are sets of a size 2 and not directed pairs.

Symmetries from Problem Description Graphs

The last creature we want to introduce in this section is the problem description graph (PDG), that was introduced by Pochter *et al.* (2011), and later on reformulated for different purposes by Domshlak *et al.* (2012), and Shleyfman *et al.* (2015). In this work we will use the definition of PDG for the FDR planning tasks. It is important to point out, that this structure has no direct use in the work below. However, since we want to illustrate some of our claims by graphic examples, PDG becomes quite helpful, since in contrast to structural symmetries PDG is a graph, and hence can be presented in a picture.

Definition 1. *Let Π be a FDR planning task. The* ***problem description graph*** *(PDG) of Π is the colored directed graph $\langle N, E \rangle$ with nodes*

$$N = N_{\mathcal{V}} \cup \bigcup_{v \in \mathcal{V}} N_{\mathcal{D}(v)} \cup N_{\mathcal{A}}$$

where $N_{\mathcal{V}} = \{n_v \mid v \in \mathcal{V}\}$, $N_{\mathcal{D}(v)} = \{n_{\langle v,d \rangle} \mid d \in \mathcal{D}v\}$, and $N_{\mathcal{A}} = \{n_a \mid a \in \mathcal{A}\}$; node colors

$$col(n) = \begin{cases} 0 & \text{if } n \in N_{\mathcal{V}} \\ 1 & \text{if } n \in \bigcup_{v \in \mathcal{V}} N_{\mathcal{D}(v)} \text{ and } \langle v, d \rangle \in G \\ 2 & \text{if } n \in \bigcup_{v \in \mathcal{V}} N_{\mathcal{D}(v)} \text{ and } \langle v, d \rangle \notin G \\ 3 + \mathsf{cost}(a) & \text{if } n_a \in N_{\mathcal{A}} \end{cases}$$

and edges

$$E = \bigcup_{v \in \mathcal{V}} E^v \cup \bigcup_{a \in \mathcal{A}} E_a^{\mathsf{pre}} \cup E_a^{\mathsf{eff}}$$

where $E^v = \{(n_v, n_{\langle v,d \rangle}) \mid d \in \mathcal{D}(v)\}$, $E_a^{\mathsf{pre}} = \{(n_a, n_{\langle v,d \rangle}) \mid \langle v, d \rangle \in \mathsf{pre}(a)\}$, and $E_a^{\mathsf{eff}} = \{(n_{\langle v,d \rangle}, n_a) \mid \langle v, d \rangle \in \mathsf{eff}(a)\}\}$.

In their work, Pochter *et al.* (2011) observed, that *PDG symmetry* is a symmetry of $\mathcal{T}_{\Pi}$ that is induced by a graph automorphism of the PDG of Π. Shleyfman *et al.* (2015), in turn, showed that every structural symmetry of Π corresponds to a PDG symmetry of Π in the sense that they induce the same transition graph symmetry. In what follows, we will denote by $Aut(\Pi)$ the automorphism group of PDG of a task Π. The illustrative examples of planning tasks will also be presented via PDGs.

Complexity

In this section we aim to prove that for each undirected graph one may construct a planning task with a similar automorphism group. Surprisingly, not much have been said about complexity of computation of such a group. We show a simple reduction that will prove that this computation is at least GI-hard.

The graph isomorphism problem (GI) is a well-known problem, that gave its name to a whole complexity class. This problem is a decision problem of determining whether two finite graphs are isomorphic. Other well know problem is the graph automorphism problem, that is a problem of testing whether a graph has a nontrivial automorphism. And this is at least as hard as solving the decision problem of either automorphism group of a given graph is trivial or not. The graph automorphism problem is polynomial-time many-one reducible to the graph isomorphism problem (Mathon 1979) (the converse reduction is unknown). Thus, given the reduction, we can say that computing the automorphism group of a given planning task is at least GI-hard. The latest result by Babai (2015) claims (most probably rightfully) that GI can be solved in quasipolynomial time, i.e. in $\exp((\log n)^{O(1)})$. The best previous bound stood on $O(\exp(\sqrt{n \log n}))$ (Babai and Luks 1983).

Reduction to bounded variable domains

While discussing relations between the automorphism groups of different structures, we first should introduce the notation of mapping and comparing these groups. In this section we will rely mostly on the basic definitions of the group theory taken from the book "Topics in algebra" (Herstein 1975). Let us start with some useful mappings:

Definition 2. *Let G and G' be groups:*

1. *and let $f : G \mapsto G'$ be a mapping that satisfies $f(ab) = f(a)f(b)$ for all $a, b \in G$, then f is a **homomorphism** of G to G'.*
2. *If f is a bijection, it is called an **isomorphism**, this is denoted by $G \cong G'$, or simply $G = G'$.*
3. *If f is a injection, then there exist a subgroup $H \leq G$, s. t. $H \cong G'$. In this case we will write simply $G \leq G'$.*

As we mentioned before, Pochter *et al.* (2011) introduced a method for deduction of the automorphism group for an FDR representation of a planning task using PDG. While this representation is easy to visualize and understand, it is a bit inconvenient as an algebraic model of representation. On the other hand, structural symmetries while having a much simpler definition, lack the graphical appeal. In the proof which follows, we want to establish a connection between the vertexes of a given undirected graph and the variables of the planning task constructed by our reduction. Thus, as the middle ground, we chose to conduct our proof using causal graphs, since they are both simple in representation and, once again, have an allure of being graphs. Unfortunately, as the next statement shows, there is no straightforward subgroup relation between the automorphism group of the planning task and the automorphism group of its causal graph.

Observation 1. *There exist a planning task Π, s. t. $Aut(\Pi) \nleq Aut(CG(\Pi))$ and $Aut(CG(\Pi)) \nleq Aut(\Pi)$.*

Proof. Let Π be a planning task with variables $\mathcal{V}$ and actions $\mathcal{A}$. Consider a set of variables $\mathcal{V} = \{v, u\}$, where $\mathcal{D}(v) = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{D}(u) = \{u_1, u_2\}$. Let $\mathcal{A}$ be a set of actions with single precondition and single effect. To ease the writing and reading of this example, we will use the following notation $a_{x_i \to y_j} := \langle \{\langle x, x_i \rangle\}, \{\langle y, y_j \rangle\} \rangle$. Thus, for example the action $a_{v_1 \to v_2} = \langle \{\langle v, v_1 \rangle\}, \{\langle v, v_2 \rangle\} \rangle$. Now, let us define a set of actions of Π, $\mathcal{A} = \{a_{v_1 \to v_2}, a_{v_2 \to v_3}, a_{v_3 \to v_1}, a_{v_1 \to v_4}, a_{v_2 \to v_4}, a_{v_3 \to v_4}, a_{v_4 \to u_1}, a_{u_1 \to u_2}, a_{u_2 \to v_4}\}$. Since we don't want our planning task to be redundant we will set $G = \{\langle v, v_4 \rangle\}$. It is easy to check that $Aut(\Pi)$ is generated by the cycle $(\langle v, v_1 \rangle, \langle v, v_2 \rangle, \langle v, v_3 \rangle)$, i. e. for some $\sigma \in Aut(\Pi)$ holds that $\sigma(\langle v, v_1 \rangle) = \langle v, v_2 \rangle$, $\sigma(\langle v, v_2 \rangle) = \langle v, v_3 \rangle$, and $\sigma^3 = id_\Pi$ to complete the cycle, and that σ is fixed on all other facts. Thus, $Aut(\Pi) \cong \mathbb{Z}_3$, and cyclic group of order 3. The causal graph and PDG of Π are depicted in Figure 1.

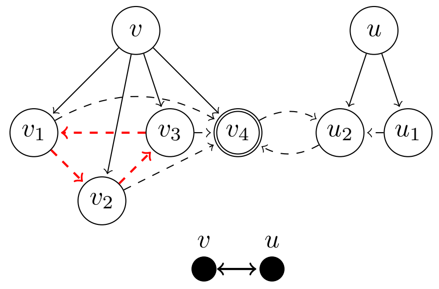

Figure 1: Illustration for Observation 1: The graph with the white nodes represents the PDG of the task described in the Observation in question. Since preconditions and effects of each action are single-valued, we annotated them via dashed arrows. The goal fact is denoted by double circle. Here it is easy to see, that the automorphism group of the PGD is generated by the cycle (v_1, v_2, v_3) (red, dashed arrows). The filled dots represent the causal graph of the same task.

On the other hand, the causal graph of task Π is $\langle N = \{v, u\}, \mathcal{E} = \{(v, u), (u, v)\} \rangle$, and has the automorphism group that is isomorphic to $\mathbb{Z}_2$. Hence, since both $\mathbb{Z}_3, \mathbb{Z}_2$ have no non-trivial subgroups the claim holds. $\square$

Since we still want to embed $Aut(\Pi)$ into $Aut(CG(\Pi))$, we will need the following Definitions and Theorem (once again taken from the "Topics in algebra" (Herstein 1975)).

Definition 3. *Let H be a subgroup of G, and let x be an element in G.*

1. *The set of elements $Hx = \{hx \mid h \in H\}$ is called a **right coset** of H. A **left coset** defined similarly.*
2. *If for every $x \in G$ holds that $Hx = xH$, H is called a **normal subgroup**.*
3. *Let H be a normal subgroup of a G. The set ${}^{G}/_{H} := \{xH \mid x \in G\}$ of all left cosets forms a **quotient group** of G modulo H.*

Below we give a short reminder on the definition of group homomorphism.

Definition 4. *Let H and G be two groups. We say that the map $\phi : G \to H$ is a **group homomorphism** (later on referred as homomorphism), if for all $g_1, g_2 \in G$ it holds that $\phi(g_1)\phi(g_2) = \phi(g_1 g_2)$.*

*The **kernel** of the map ϕ, denoted by $ker(\phi)$, is the set $\phi^{-1}(id_H)$.*

*We say that ϕ is an **isomorphism**, if in addition to the mentioned above it is also a bijection.*

Note, that it is easy to prove that $ker(\phi) \leq G$ and $\phi(G) \leq H$ are both subgroups. To establish additional relationships between homomorphisms, quotients, and subgroups we will need the following theorem by Noether (1927).

Theorem 1 (First isomorphism theorem)**.** *Let G and H be groups, and let $\phi : G \to H$ be a homomorphism. Then:*

1. *The kernel of ϕ is a normal subgroup of G,*
2. *The image of ϕ is a subgroup of H, and*
3. *The image of ϕ is isomorphic to the quotient group ${}^{G}/_{ker(\phi)}$.*

In particular, if ϕ is surjective then H is isomorphic to ${}^{G}/_{ker(\phi)}$.

Intuitively, the next Lemma shows that if we strip from $Aut(\Pi)$ all the automorphism that do not affect the tasks variables, the resulted subgroup can be embedded into the automorphism group of $CG(\Pi)$.

Lemma 1. *Let $\phi : Aut(\Pi) \mapsto Aut(CG(\Pi))$ be a map s.t. for each $\sigma \in Aut(\Pi) : \phi(\sigma) = \sigma_{\mathcal{V}}$, where $\sigma_{\mathcal{V}}$ is σ restricted to $\mathcal{V}$.*

Then, ϕ is a homomorphism, and ${}^{Aut(\Pi)}/_{ker(\phi)} \leq Aut(CG(\Pi))$.

Proof. Once again, let Π be a planning task with variables $\mathcal{V}$ and actions $\mathcal{A}$. First, let us prove that $\sigma_{\mathcal{V}}$ is an automorphism. Let $(u, v) \in \mathcal{E}$ be an edge in $CG(\Pi)$. Hence, exists $a \in \mathcal{A}$, s.t. $u \in vars(\mathsf{pre}(a)) \cup vars(\mathsf{eff}(a))$ and $v \in vars(\mathsf{eff}(a))$. And therefore, for each $\sigma \in Aut(\Pi)$ it holds that $\sigma(u) \in vars(\mathsf{pre}(\sigma(a))) \cup vars(\mathsf{eff}(\sigma(a)))$ and $\sigma(v) \in vars(\mathsf{eff}(\sigma(a)))$. From which follows that $(\sigma_{\mathcal{V}}(u), \sigma_{\mathcal{V}}(v)) \in \mathcal{E}$. The converse is true, since each σ^{-1} is also an automorphism.

Second, ϕ is a homeomorphism, since for each $\sigma, \sigma' \in Aut(\Pi)$, it holds that $\phi(\sigma)\phi(\sigma') = \sigma_{\mathcal{V}}\sigma'_{\mathcal{V}} = (\sigma\sigma')_{\mathcal{V}} = \phi(\sigma\sigma')$, given that ϕ is a restriction to variables.

Now, $ker(\phi) = \{\sigma \in Aut(\Pi) \mid \sigma = id_{\mathcal{V}}\}$, and by the first isomorphism theorem it holds that ${}^{Aut(\Pi)}/_{ker(\phi)} = \phi(Aut(\Pi)) \leq Aut(CG(\Pi))$. □

Following the intuition of Lemma 1, in the Theorem below we construct a planning task that has no "inner" automorphism. The automorphism group of such task should be isomorphic to the automorphism groups of its causal graph. The theorem is the main result of this section.

Theorem 2. *Let $\mathcal{G}$ be a directed graph without loops, then there exists a planning task Π, s.t. $\mathcal{G} = CG(\Pi)$, $Aut(\mathcal{G}) = Aut(\Pi)$.*

Proof. In this proof, given a directed graph $\mathcal{G} = \langle N, E\rangle$, we should construct a planning task Π that satisfies the conditions of the Theorem. First, it's clear that vertex $x \in N$ should correspond a variable $v \in \mathcal{V}$. Now, since we would like to use Lemma 1, the kernel of the homomorphism ϕ should be trivial. Thus, we set $\mathcal{D}v = \{T, F\}$, and add an action $a_{v:F\to v:T} := \langle\{\langle v, F\rangle\}, \{\langle v, T\rangle\}\rangle$, s. t. for each $\sigma \in Aut(\Pi)$ holds $\sigma(\langle v, F\rangle) \neq \langle v, T\rangle$. For each $(x, y) \in E$, let v and u be the corresponding variables in $\mathcal{V}$. To ensure that $\mathcal{G} = CG(\Pi)$, we add a unique action $a_{u:F\to v:F} := \langle\{\langle u, F\rangle\}, \{\langle v, F\rangle\}\rangle$, which, in turn, assures that if $\sigma(a_{u:F\to v:F}) \neq a_{u:F\to v:F}$, then either $\sigma(v) \neq v$ or $\sigma(u) \neq u$. In addition, we need to specify an initial state, and a goal description. Let those two be full assignments $I := \{\langle v, F\rangle \mid v \in \mathcal{V}\}$ and $G := \{\langle v, T\rangle \mid v \in \mathcal{V}\}$. Since, by construction of Π, σ never maps T to F, this leaves the automorphism group $Aut(\Pi)$ unchanged. To summarize, the constructed planning task $\Pi = \langle\mathcal{V}, \mathcal{A}, I, G\rangle$ looks as follows:

1. $\mathcal{V} = \{v \mid v \in N\}$, with $\mathcal{D}v = \{T, F\}$ for each v,
2. $\mathcal{A} = \{a_{v:F\to v:T} \mid v \in V\} \cup \{a_{v:F\to u:F} \mid (v, u) \in E\}$,
3. $I = \{\langle v, F\rangle \mid v \in \mathcal{V}\}$, and
4. $G = \{\langle v, T\rangle \mid v \in \mathcal{V}\}$.

Since the algebraic mapping can be hard to imagine, the PDG structure of edge (u, v) is depicted in Figure 2.

Now let ϕ be a homomorphism as defined is Lemma 1. By construction of Π, ϕ is surjective and $ker(\phi) = id_{\mathcal{G}}$, thus $Aut(\mathcal{G}) = Aut(\Pi)$. □

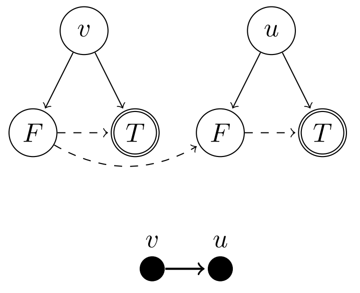

Figure 2: Illustration of a mapping of a single edge in a graph for Theorem 2: Once again, the graph with the white nodes represents the PDG of and edge (v, u) (depicted by filled nodes). Here it easy to see that there no "inner" symmetries, and the planning variable $(v, \mathcal{D}(v))$ can be mapped into a planning variable $(u, \mathcal{D}(u))$ exactly in one way.

Now what is left to show, is that there is an automorphism group preserving reduction from a undirected graphs to directed graphs.

Proposition 1. *Let $\mathcal{G}$ be a undirected graph, then there exists directed graph $\mathbb{G}$, s.t. $Aut(\mathcal{G}) = Aut(\mathbb{G})$.*

The proof of this statement is not new, but we will use it later on to show that even special cases of planning tasks are difficult to solve, in the sense of finding the automorphism group.

Proof. Let $\mathcal{G} = \langle N, E\rangle$ be an undirected graph. Let us define $\mathbb{G} = \langle\mathcal{V}, \mathcal{E}\rangle$ as follows:

1. $\mathcal{V} := N \cup E$, and
2. $\mathcal{E} := \{(e, x), (e, y) \mid e = \{x, y\} \in E\}$.

Note, that for the vertices in $\mathcal{V}$ for $x \in N$ and $e \in E$, $deg_{out}(x) = deg_{in}(e) = 0$. The graphic example of this construction can be seen if Figure 3. Hence, for each $\sigma \in Aut(\mathbb{G})$ holds that $\sigma(N) = N$ and $\sigma(E) = E$, where N and E are both sets of vertices in $\mathbb{G}$. Moreover, for each edge $e = \{x, y\}$ in E correspond to edges $(e, x), (e, y)$ in $\mathcal{G}$. Thus, for $\sigma \in Aut(\mathbb{G})$, $e = \{x, y\} \in \mathcal{E}$ iff $\sigma(e) = \{\sigma(x), \sigma(y)\} \in \mathcal{E}$ which corresponds to $(e, x), (e, y) \in \mathcal{E}$ iff $(\sigma(e), \sigma(x)), (\sigma(e), \sigma(y)) \in \mathcal{E}$. Using this, we will define $\phi : Aut(\mathcal{G}) \to Aut(\mathbb{G})$:

$$\phi(\sigma) = \begin{cases} \sigma(x) & \text{if } x \in N, \\ \sigma(e) = \{\sigma(x), \sigma(y)\} & \text{if } e = \{x, y\} \in E. \end{cases}$$

Now, to prove that ϕ is an isomorphism we need to prove that ϕ is surjection, and that $ker(\phi) = \{id_{\mathcal{G}}\}$. First, for each $\tau \in Aut(\mathbb{G})$, $\phi^{-1}(\tau) = \tau|_N \in Aut(\mathcal{G})$. Second, $\phi^{-1}(id_{\mathbb{G}}) = id_{\mathbb{G}}|_N = id_{\mathcal{G}}$. Thus, by the first isomorphism theorem it holds that ${}^{Aut(\mathcal{G})}/_{\{id_{\mathcal{G}}\}} = Aut(\mathcal{G}) = Aut(\mathbb{G})$. □

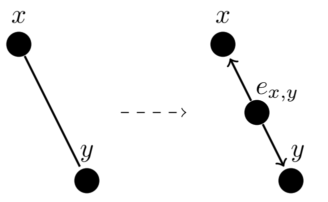

Figure 3: Illustration of a mapping of a single edge in a graph for Proposition 1: edge $e_{x,y} = \{x, y\} \in E$ is mapped to a vertex $e_{x,y} \in \mathcal{V}$ and two edges $(e, x), (e, y) \in \mathcal{E}$.

The next Corollary is the immediate consequence of Proposition 1 and Theorem 2.

Corollary 1. *Given a planing task* Π*, computing* $Aut(\Pi)$ *is as equivalent to computing* $Aut(\mathcal{G})$ *for some undirected graph* $\mathcal{G}$*.*

Proof of Proposition 1 also show thats even planning tasks that have a bipartite one-way directed causal graphs (fork decomposition by Katz and Domshlak (2008)) may have an arbitrary finite automorphism group, which follows from the next theorem proven by Frucht (1949)

Theorem 3 (Frucht's theorem)**.** *Every finite group is the automorphism group of a finite undirected graph.*

Reduction to singe variable domain

As for now, we have seen that given an undirected graph $\mathcal{G}$ we can construct a planning task Π with the same automorphism group, and where each vertex in the graph $\mathcal{G}$ corresponds with a bounded variable in a task Π. In this section we show, that if we remove the bounded domains condition, only one variable for such a task will suffice.

Zemlyachenko *et al.* (1985) showed that finding an isomorphism of a connected graph is a GI-complete problem. Therefore, to prevent the task from being reducible via standard preprocessing we will take the graph $\mathcal{G}$ to be a connected undirected graph.

Proposition 2. *Let* $\mathcal{G}$ *be a connected undirected graph, then there exists planning task* $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$*, s.t.* $Aut(\mathcal{G}) = Aut(\Pi)$ *and* $|\mathcal{V}| = 1$*.*

Proof. Let $\langle N, E \rangle$ be the vertices and edges of $\mathcal{G}$, correspondingly, and let $\mathcal{V} = v$ be single variable in the task Π. We will define the domain on v to be $\mathcal{D}(v) := \{v_x \mid x \in N\} \cup \{v_g\}$, where v_g is the goal value of v ($G := \{\langle v, v_g \rangle\}$). Now, since the structural symmetries ignore the initial state, all is left to do is to define the actions of this task. Since we have only one variable we will use the following notation $a_{v_x \to v_y} := \langle \{\langle v, v_x \rangle\}, \{\langle v, v_y \rangle\} \rangle$. The actions $\mathcal{A}$ of our task will be divided into two sets:

1. $\mathcal{A}_E := \{a_{v_x \to v_y}, a_{v_y \to v_x} \mid e = \{x, y\} \in E\}$, and
2. $\mathcal{A}_g := \{a_{v_x \to v_g} \mid x \in N\}$.

Now, let us look at the map $\psi : N \to \{\langle v, d \rangle \mid d \in \mathcal{D}(v)\}$, by construction ψ is injective. Therefore the map $\phi : Aut(\mathcal{G}) \to Aut(\Pi)$:

$$\phi(\sigma) = \begin{cases} \sigma(\psi(x)) & \text{if } x \in N, \\ \langle v, v_g \rangle & \text{otherwise} \end{cases}$$

is also injective, since it's easy to see that ψ preserves the relation on the edges, $\psi : E \to \mathcal{A}_E$ set-wise. The injection follows from the fact that $\langle v, v_g \rangle$ is a unique goal fact that can be mapped by σ only upon itself, and $\psi : N \to \mathcal{A}_g$ is a bijection. Thus we get the desired $Aut(\mathcal{G}) = Aut(\Pi)$. □

Group Presentation

In this section we will discuss the presentation of symmetry groups, and show that for some planning domains this task may be quite difficult.

Most of the tools for computing automorphism groups, such as *Bliss* (Junttila and Kaski 2007), *nauty* (McKay and Piperno 2014), and *saucy* (Darga *et al.* 2008), report the set of generators required to produce the $Aut(G)$ automorphism group of a given graph G. Thus, in some works, the authors (Sievers *et al.* 2015; 2017) chose to report this numbers for each group, or even for each planning domain in the experimental benchmarks. In this subsection we will show the faults in this approach. To do so, we will need some standard definitions:

Definition 5. *Let* G *be a group. We say that* G *has a* ***presentation*** $\langle S \mid R \rangle$*, where* S *is a set of* ***generators*** *so that every element of the group can be written as a product of powers of some of these generators, and* R *is a set of* ***relations*** *among those generators.*

Let $F(S)$ *be a* ***free group*** *on* S *(all finite words of* S *with the relation* $suu^{-1}t = st$*). The set of relations* R *is the subset of* $F(S)$*.*

The group G *is said to have the above presentation if it is isomorphic to the quotient of* $F(S)$ *by the minimal normal subgroup that contains the set* R*.*

We say that presentation $\langle S \mid R \rangle$ *of group* G *is* ***irreducible*** *if for no* $S' \subsetneq S$ *holds that* G *isomorphic to* $\langle S' \mid R|_{S'} \rangle$*.*

From the First isomorphism theorem follows that every finite group has a presentation. And as corollary of this statement, easily obtained, that every finite group is finitely generated, since S can be taken to be G itself. To get a better grip on this definition we will present an couple of examples, that will be use further in this section.

Example 1. *The **cyclic group** is a group generated by a single element. The group $\mathcal{C}_k$ can be presented as $\langle S \mid R\rangle$ where:*

- $S := \{\sigma\}$*;*
- $R := \{\sigma^k\}$.

It is easy to see that for a given $k \in \mathbb{N}$, it holds that $|\mathcal{C}_k| = k$, and only one generator. Example 1 provides us the fact that the number of generators does not ensure the upper bound on the size of the group. To calculate the lower bound we will prove the following lemma[2]:

Lemma 2. *Let G be a group with presentation $\langle S \mid R\rangle$, and let be $S = \{g_1, \dots, g_n\}$ set of n irreducible generators. Then, $|G| \geq 2^n$.*

Proof. Let G^m be a subgroup of G that has a set of generators $\{g_1, \dots g_m\}$, for $1 \leq m < n$. Since the set S is irreducible with respect to G, every subset of S is also irreducible with respect to the subgroup of G it generates, thus $g_{m+1} \notin G^m$. Therefore, G^{m+1} has at least two cosets $e\,G^m = G^m$ and $g_{m+1}G^m$. By definition of cosets it holds that $G^m \cap g_{m+1}G^m = \emptyset$. Which leads to $|G^{m+1}| \geq 2|G^m|$. Thus, by induction on m we have that $G^n \geq 2^n$. □

The equality for the Lemma above is achieved on the group $\mathcal{C}_2^n \cong \mathbb{Z}_2^n = \prod_{i=1}^n \mathbb{Z}_2$. Giving us that the amount of elements (*order*) of a finite group, is at least exponential in the size of group generators.

To show that group structure is not defined by the number of generators we will need at least one another group, that is not cyclic. For that we will define the symmetric group. Note, that in literature symmetry group is often used as a synonym to the automorphism group of some mathematical object, symmetric group, on the other hand, is a group of all permutations of some n identical objects.

Example 2. *The **symmetric group** $\mathcal{S}_n$ on a finite set of n symbols is the group whose elements are all the permutations on n distinct symbols. The group $\mathcal{S}_n$ can be written as $\langle S \mid R\rangle$ where:*

- $S := \{\sigma_i | i \in [n-1]\}$*;*
- $R := \{\sigma_i^2 | i \in [n-1]\} \cup \{\sigma_i\sigma_j\sigma_i^{-1}\sigma_j^{-1} | i \neq j \pm 1\} \cup \{(\sigma_i\sigma_j)^3 | i,j \in [n-1]\}$

It's important to point out (and easy to check) that $\mathcal{S}_n$ has $n!$ elements. Using the cyclic notation, each element in the presentation can be written as $\sigma_i = (i, i+1)$, which means that σ_i maps the element i to element $i+1$, element $i+1$ is mapped to i, and other elements are mapped to itself. This presentation is irreducible (Alperin and Bell 1995), meaning that non of the permutations can be excluded from the set S, where $|S| = n-1$.

As we showed a bit earlier in this section the group $\mathcal{C}_2^n$ also has n generators, each of order two[3]. However, it is obvious that $\mathcal{C}_2^n \ncong \mathcal{S}_{n+1}$, since $2^n \neq (n+1)!$, for $n > 1$.

[2]This result is well known in group theory, but unfortunately we haven't found citing source.

[3]Order of an element g is the minimal number m s.t. $g^m = e$.

By this example, reporting the number of generator per domain, or even for a specific group (even with order of these generators) is not very informative, since this numbers reports us almost nothing on the size and structure of the group. Note that the group $\langle a, b \,|\, a^2, b^2\rangle$ is of an infinite size (as an intuition, consider the following elements $a, ab, aba, abab, \dots$).

Another way to write the cyclic group $\mathcal{S}_n$, may be given with only two cyclic generators $(1,2)$ and $(2,\dots,n)$, which is also irreducible (Alperin and Bell 1995). Note that in the first cyclic presentation of $\mathcal{S}_n$ each generator has an order of 2, and in the second the first cycle is of order 2, and the second one is of order $n-1$. As we can see, each group may have more than one presentation. And calculating the minimal number of these generators per group is known to be at most $O(\log^2\ n)$ space (Arvind and Torán 2006).

Conclusion

An automorphism group of a planning task can be seen as permutation on objects involved in this task, and thus it constitutes a subgroup of some symmetric group. In this sense, our results coincide with the famous Cayley's theorem (Herstein 1975).

Theorem 4 (Cayley's theorem)**.** *Every group G is isomorphic to a subgroup of the symmetric group acting on G.*

The obvious corollary of this theorem is that, every finite group is isomorphic to a subgroup of the symmetric group.

One may have an intuition, that since planning tasks are constructed from objects, the symmetry groups of such tasks can be subjects to a special treatment. This is apparently false in the general case. Thus, we want to conclude this unoptimistic paper by a quote from the book "Groups and representations" by Alperin and Bell (1995): "in general the fact that finite groups are embedded in symmetric groups has not influenced the methods used to study finite groups". Unfortunately, by reduction we showed in the previous Section, this statement also holds for the automorphism groups of the tasks in classical planning.

References

J.L. Alperin and R.B. Bell. *Groups and representations*. Graduate texts in mathematics. Springer, 1995.

Vikraman Arvind and Jacobo Torán. The complexity of quasigroup isomorphism and the minimum generating set problem. In *ISAAC*, 2006.

László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 171–183, 1983.

László Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015.

Fahiem Bacchus and Qiang Yang. Downward refinement and the efficiency of hierarchical problem solving. *AIJ*, 71(1):43–100, 1994.

Christer Bäckström and Bernhard Nebel. Complexity results for SAS$^+$ planning. *Computational Intelligence*, 11(4):625–655, 1995.

Paul T. Darga, Karem A. Sakallah, and Igor L. Markov. Faster symmetry discovery using sparsity of symmetries. In *Proceedings of the 45th Annual Design Automation Conference*, DAC '08, pages 149–154, New York, NY, USA, 2008. ACM.

Carmel Domshlak and Ronen I. Brafman. Structure and complexity in planning with unary operators. In Malik Ghallab, Joachim Hertzberg, and Paolo Traverso, editors, *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, April 23-27, 2002, Toulouse, France*, pages 34–43. AAAI Press, 2002.

Carmel Domshlak, Michael Katz, and Alexander Shleyfman. Enhanced symmetry breaking in cost-optimal planning as forward search. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*. AAAI Press, 2012.

E. Allen Emerson and A. Prasad Sistla. Symmetry and model-checking. 9(1/2):105–131, 1996.

Maria Fox and Derek Long. The detection and exploitation of symmetry in planning problems. In Thomas Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 1999)*, pages 956–961. Morgan Kaufmann, 1999.

Robert Frucht. Graphs of degree three with a given abstract group. 1(??):365–378, 1949.

Malte Helmert. The Fast Downward planning system. *JAIR*, 26:191–246, 2006.

Malte Helmert. Landmark heuristics for the pancake problem. In Ariel Felner and Nathan Sturtevant, editors, *Proceedings of the Third Annual Symposium on Combinatorial Search (SoCS 2010)*, pages 109–110. AAAI Press, 2010.

I.N. Herstein. *Topics in algebra*. Xerox College Pub., 1975.

Tommi Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In David Applegate, Gerth Stølting Brodal, Daniel Panario, and Robert Sedgewick, editors, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics*, pages 135–149. SIAM, 2007.

Michael Katz and Carmel Domshlak. Structural patterns heuristics via fork decomposition. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008)*, pages 182–189. AAAI Press, 2008.

Craig A. Knoblock. Automatically generating abstractions for planning. *AIJ*, 68(2):243–302, 1994.

R. Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8:131–132, 1979.

Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, {II}. *Journal of Symbolic Computation*, 60(0):94 – 112, 2014.

E. Noether. Abstrakter aufbau der idealtheorie in algebraischen zahl- und funktionenkörpern. *Mathematische Annalen*, 96:26–61, 1927.

Nir Pochter, Aviv Zohar, and Jeffrey S. Rosenschein. Exploiting problem symmetries in state-based planners. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the 25th National Conference of the American Association for Artificial Intelligence (AAAI'11)*, San Francisco, CA, USA, July 2011. AAAI Press.

J. Rintanen. Symmetry reduction for SAT representations of transition systems. In Enrico Giunchiglia, Nicola Muscettola, and Dana Nau, editors, *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS'03)*, pages 32–41, Trento, Italy, 2003.

Alexander Shleyfman, Michael Katz, Malte Helmert, Silvan Sievers, and Martin Wehrle. Heuristics and symmetries in classical planning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 3371–3377. AAAI Press, January 2015.

Silvan Sievers, Martin Wehrle, Malte Helmert, and Michael Katz. An empirical case study on symmetry handling in cost-optimal planning as heuristic search. In Steffen Hölldobler, Markus Krötzsch, Rafael Peñaloza, and Sebastian Rudolph, editors, *KI 2015: Advances in Artificial Intelligence - 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings*, volume 9324 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2015.

Silvan Sievers, Gabriele Röger, Martin Wehrle, and Michael Katz. Structural symmetries of the lifted representation of classical planning tasks. In *HSDIP 2017*, 2017.

Peter Starke. Reachability analysis of petri nets using symmetries. *Journal of Mathematical Modelling and Simulation in Systems Analysis*, 8(4/5):293–304, 1991.

Martin Wehrle, Malte Helmert, Alexander Shleyfman, and Michael Katz. Integrating partial order reduction and symmetry elimination for cost-optimal classical planning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1712–1718, 2015.

V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, May 1985.