Technical Report

IRI-TR-17-05

Constraint Symmetries

Pol Mestres
Vicente Ruiz de Angulo

August 2017

Institut de Robòtica i Informàtica Industrial

Abstract

Constraint Symmetries have been suggested as an appropriate subgroup of solution symmetries that can be more easily found than the complete group of symmetries. The concept is analyzed, and it becomes clear that a knowledge of the problem as deep as for finding the whole group is required to find that subset. Indeed, we show that the Microstructure Complement in most cases only finds a small fraction of the subgroup of Constraint Symmetries. Moreover, not all the symmetries it finds are truly Constraint Symmetries. We show also that in the context of point symmetries (as opposed to literal symmetries), the subgroup of Constraint Symmetries coincides with the whole solution symmetry group.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)
http://www.iri.upc.edu

Corresponding author:

V. Ruiz de Angulo
tel: +34 93 401 7337
ruiz@iri.upc.edu
http://www.iri.upc.edu/staff/ruiz

1 Introduction

The solution symmetries of a Constraint Satisfaction Problem ([1]) (CSP) have been defined as permutations of literals that preserve the set of solutions [2]. These are the more general type of symmetries. Important subtypes are value symmetries, which permute only literals of the same variable, variable symmetries, which permute only literals with the same value, and compositional symmetries, compositions of value and variable symmetries. The whole set of solution symmetries (in the following also named simply symmetries) is difficult to identify. Indeed, the knowledge of the set of solutions is required to determine it. This means that, in order to find all symmetries, we have to first solve the problem.

Therefore, for the practical purpose of symmetry detection, restricted subsets are considered. The subset of solution symmetries that appears as more intuitive for the practical purpose of symmetry detection is the one in which the constraints are permuted. These symmetries have been used often in the context of continuous constraint problems [7, 8]. But in the discrete domain, the subset that has been fixed almost as a standard is the set of Constraint Symmetries (CS), which is the set of literal permutations that preserve the constraints. Gent et al [4] provide a suitable interpretation of this definition, as a permutation that permutes the set of tuples of all constraints as a whole.

On the other hand, the tools usually proposed to perform practical symmetry detection are graphs whose automorphisms ([5]) are a subgroup of the symmetries of the CSP. One of these graphs is the Microstructure Complement (MC) of the CSP, whose automorphisms have been identified with Constraints Symmetries, resulting in an alternative, more formal definition of them. In these paper we analyze both definitions, elucidating their equivalence. We also investigate which types of symmetries are found by the MC and show other graphs whose automorphisms are arelevant subset of solution symmetries. Finally we extend the concept of Constraint Symmetry to a different class of symmetries not based on literal permutations, but on domain point permutations.

2 Solution symmetries

A CSP is a tuple $P =< X, D, C >$ where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables, $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains where D_i specifies possible values for variable $x_i \in X$, and $C = \{C_1, \ldots, C_t\}$ is a set of constraints. Each constraint is a predicate or boolean function on a subset of X, called the scope of the constraint. We denote by $\bar{C}_i$ the negation of C_i. A pair (x_i, v) where $v \in D_i$ is called a literal or an assignment to a variable. L will denote the set of literals. An assignment is a set of literals of different variables. If the assignment contains n elements it is a complete assignment or point. Otherwise it is a Partial Assignment (PA). The largest cardinality among the problem constraints scopes is the arity of the problem.

A solution to a CSP is a complete assignment such that its restriction to the scope of each constraint satisfies the constraint. Therefore C_i can be regarded as a list of PAs or tuples in the scope of the constraint allowed by it, and $\bar{C}_i$ a the list of tuples of that scope disallowed by the constraint. A solution symmetry (or simply a symmetry) of a CSP is a permutation of L that maps solutions to solutions.

A variable symmetry is defined as a symmetry s such that there exists a permutation θ of $\{1, \ldots, n\}$, $(x_i, a)^s = (x_{i^\theta}, a)$. A value symmetry s has the form $(x_i, a)^s = (x_i, a^{\theta_i})$, where θ_i is a permutation of D_i. Then, a compositional symmetry is the result of composing a variable and a value symmetry, i.e, a symmetry s that can be expressed as $(x_i, a)^s = (x_{i^\theta}, a^{\theta_i})$, where θ and θ_i are permutations of $\{1, \ldots, n\}$ and D_i, respectively. Finally, non-compositional ones are those that cannot be obtained by composing variable and value symmetries.

Cohen et al. [2] showed that the symmetry group of literal symmetries of a CSP is the automorphism group of the k-nogood graph (KNG) of a k-ary problem, whose set of nodes is the set of possible literals and whose set of edges is the set of all m-ary nogoods for all $m \leq k$. A k-nogod here must be understood as a set of literals that cannot be extended to a solution.

Consider the graph (which we will call the solution graph) whose vertices are the literals and whose edges are the sets of literals forming a solution. It is easy to see that the group of automorphisms of this graph is the set of solution symmetries.

It might be surprising the fact that two very different graphs give exactly the same group of automorphisms (the k-nogood and the solution graph), but in fact it can be proven that if we consider the complement of the solution graph, it's just an extension of the k-nogood graph such that the added edges were already permuted in an automorphism of the k-nogood, so the group of automorphims is the same.

Lemma. The complement of the solution graph has the same automorphims as the KNG.

Proof. The complement of the solution graph is nothing but an extension of the k-nogood graph such that the added edges don't change the automorphism group.
The edges of the complement of the solution graph that weren't in the k-nogood graph are the following:
a) subsets of a solution of cardinality up to k.
b) sets of cardinality between $k+1$ and $n-1$
c) non-solution points.
It's clear that automorphisms of the k-nogood graph send non-edges to non-edges, so sets of type a), b) and c) are sent to sets of type a), b) and c) respectively. Thus, automorphisms of the k-nogood graph are also automorphisms of the complement solution graph.
Now let's see why every automorphism of the complement solution graph is an automorphism of the k-nogood graph. To do so it is sufficient to prove again that sets (now edges) of the form a), b) and c) are sent to sets of the form a), b) and c) respectively. Suppose π is an automorphism of the complement solution graph. If a is a set of type a), it can be extended to a solution sol. Suppose $\pi(a)$ isn't a subset of type a). Then by cardinality it's either a nogood, a forbidden assignment or an invalid assignment, but none of these sets can be a subset of $\pi(\text{sol})$, that is a solution. It's trivial to show that sets of the form b) and c) are sent to sets of the form b) and c) respectively.
Hence we have shown that automorphisms of the complement solution graph are also automorphims of the k-nogood graph.

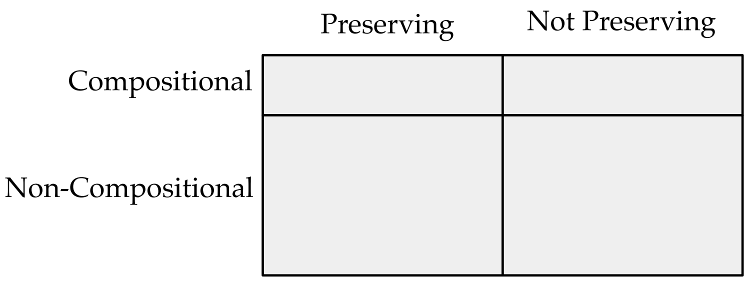

Figure 1: Subdivision of solution symmetries following two dichotomies: 1) Being constraint preserving or not (labeled Preserving). 2) Being compositional or not (labeled Compositional). All types of symmetries are found by KNG and the solution graph.

Therefore, KNG is only basically the complement of the simple solution graph. Note that the nogoods are an –extensive– way of codifying the set of solutions. Thus, in any case the knowledge of the solutions is required to identify solution symmetries.

In the following sections compositionality will reveal itsefl as a fundamental property to understand the relation between constraint preservation and detectability. For this reason, we will represent symmetries using two dichotomies following the properties of constraint preservation and compositionality (Figure 1). Note that compositional symmetries are represented as a smaller set that non-compositional ones, because compositional permutations can be shown to be a small fraction of the whole set of literal permutations. Of course, both KNG and the solution graph find all subsets in the table.

Note also that the identity permutation is compositional and preserves the constraints, so any set of symmetries with group structure should contain the up-left box. Since the set of automorphisms of a graph always has group structure, all the graphs we will consider have the compositional constraint preserving box filled.

3 Constraint symmetries

3.1 Constraint Preservation Definition

According to [2, 4] a constraint symmetry is a permutation of literals preserving the set of constraints. And following [4] this must be interpreted as a permutation of literals such that when applied to the set of tuples of all constraints, we get again the whole set of tuples of the constraints. That is, a bijection s of literals such that if $\mathbf{a} \in \bigcup_i C_i$, $s(\mathbf{a}) \in \bigcup_i C_i$. There is another way to look at the constraints: they can be also specified by the set of tuples disallowed by them. Thus, we can also define a constraint symmetry as a bijection of literals preserving the whole set of disallowed tuples. This version is more natural that the preceding one. Indeed, the set of allowed tuples of all constraints does not determine the set of solutions. Instead, the set of tuples allowed *by all constraints* do determine the solutions. The complement of this set is the set of disallowed tuples and, thus, the information about the solutions set is kept by a literal permutation preserving the set of disallowed tuples. In addition this is closer to an automorphism of the Microstructure Complement graph described next. Thus we adopt the negative version. However, nor this definition, neither the preceding one, do guarantee that such bijection is a symmetry. This is an example of a literal permutation s preserving the negative PA's of the constraints, $s(\cup_i \bar{C}_i) = \cup_i \bar{C}_i$, but not being a solution symmetry:

$$CSP = \begin{cases} Variables\ \{x, y\} \\ Domain\ \{0, 1\} \\ Constraint\ x + y = 1 \end{cases}$$
$$Permutation\ (x = 0\ \ y = 0)$$

It is perhaps because of this that [9] identifies constraint symmetry with constraint preservation, but requiring that the permutation of literals is a solution symmetry in the first place.

In sum, we take the following as Constraint Preserving Symmetry (CPS) definition and also as canonical definition of Constraint Symmetry: a *solution symmetry* s satisfying $s(\cup_i \bar{C}_i) = \cup_i \bar{C}_i$.

The following proposition shows that there exists an equivalent definition that does not impose the condition of being a symmetry in the first place:

Proposition. s is a symmetry preserving the constraints iff s is a literal permutation preserving the constraints such that $s(x)$ is a point if x is a solution.

Proof. $\Rightarrow$ This implication is obvious.
$\Leftarrow$ We prove that s must be a symmetry, that is, that s maps every solution to a solution. To begin, note that a set of literals y is a solution iff it is a point and satisfies all constraints, i.e, $a \subseteq y \Rightarrow a \notin \cup_i \bar{C}_i$. Let x be a solution. We already know that $s(x)$ is a point. To verify that $s(x)$ is a solution we have only to prove that $s(x)$ satisfies all constraints, i.e., that $a \subseteq s(x) \Rightarrow a \notin \cup_i \bar{C}_i$. Since s is a bijection , any subset of $s(x)$ is the image of a unique subset of x. Let a be an arbitrary subset of $s(x)$ and b the subset of x such that $s(b) = a$. s permutes $\cup_i \bar{C}_i$ and, by definition, s preserves the constraints, i.e., permutes $\cup_i \bar{C}_i$. That means that the image of an allowed tuple can only be an allowed tuple, i.e., $a \notin \cup_i \bar{C}_i \Rightarrow s(a) \notin \cup_i \bar{C}_i$. Since $b \subseteq x$ and x is a solution, $b \notin \cup_i \bar{C}_i$ and, therefore, $s(b) = a \notin \cup_i \bar{C}_i$. Thus, $\forall a \subseteq s(x)$, $a \notin \cup_i \bar{C}_i$, i.e., $s(x)$ satisfies the constraints and is a solution point. We conclude that s is a symmetry, and one preserving the constraints, as established in the premise.

This result has important implications and we will insist on it later.

Lemma. The set of CPS's is a group under composition.

Proof. i) closed under composition: recall that the set of solution symmetries is a group under composition and that if π, ν are constraint symmetries and $a \in \cup_i \bar{C}_i$, $\pi(a) \in \cup_i \bar{C}_i$ and $\nu(a) \in \cup_i \bar{C}_i$, so $\nu(\pi(a)) \in \cup_i \bar{C}_i$.
ii) If a permutation s is a symmetry and preserves $\cup_i \bar{C}_i$ then s^{-1} is a symmetry and also permutes $\cup_i \bar{C}_i$. iii) the identity element exists, it's the identity symmetry, which preserves $\cup_i \bar{C}_i$.

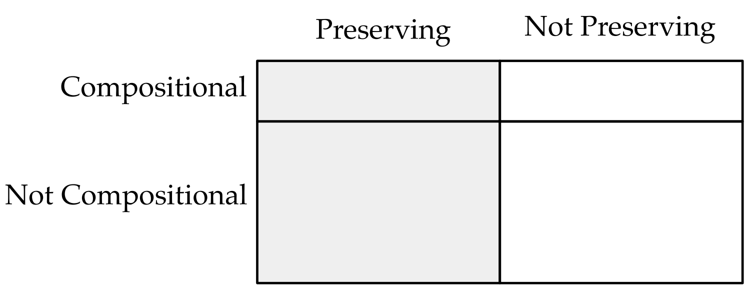

Figure 2: Symmetries found by the CSPG.

It is possible to create a graph whose automporphisms coincide with the CPS's of a CSP. That is, a graph that finds the symmetries in the first column in our table, as showed in Figure 2. Let the CPS Graph (CPSG) of a k-ary CSP be the hypergraph whose vertices are the literals and a dummy vertex d and having the following sets of hyperedges:
a) Sets of cardinality m for $m \leq k$ not forming a valid PA (i.e, such that there exist at least two vertices of the set associated to the same variable).
b) Sets of the form $\{d\} \cup \{$PA forbidden by a constraint$\}$.
c) m-ary nogoods $\forall\ m \leq k$ not forbidden by a constraint.
d) $\{d\}$.

Definition. A quasi-independent set of vertices of the CPSG is a set of vertices A containing d such that $A \backslash \{d\}$ is independent, that is, such that it only contains an edge: $\{d\}$

Lemma. Let sol be a solution of a CSP. Then s is such that sol $\cup \{d\}$ is quasi-independent.

Proof. Suppose we have a solution of the CSP, i.e, a set of n literals such that: i) no two literals have the same variable, and ii) no subset of literals is disallowed by a constraint.
By construction of the CPSG, if we add d to the set of literals, we have a quasi-independent set, because if we had an edge other than $\{d\}$ we would arrive at a contradiction. Indeed, if the edge contained d, it would also contain an assignment forbidden by a constraint, since by construction of the CPSG this is the only kind of edges in which d can be. Alternatively, if the edge doesn't contain d, then it either is a nogood (but then it couldn't be extended to a solution) or it's a set of cardinality $m \leq k$ not forming a valid PA, which also contradicts the fact that the initial set of n literals is such that no two literals have the same variable.

Remark. The converse is not true. There exist sets of $n+1$ quasi-independent vertices of the CPSG which aren't a solution. Consider for instance a problem with arity 1. Assignments of the form $x = 1, x = 2$ wouldn't have an edge, because by construction of the CPSG only invalid assignments of size 1 would have an edge (which is equivalent to saying that there are no edges of this form since clearly an assignment of size 1 is always a valid assignment). But clearly a set of vertices containing a subset of this kind would never be a solution.

Remark. If $k > 1$ the converse is true: any set of non-d vertices of a quasi-independent set of size $n+1$ of the CPSG is a solution of the CSP. Proof: We have to prove that the set of non-d vertices is such that it isn't forbidden by a constraint and there are no two vertices associated to the same variable. The former follows from the fact that there is no edge of the form $\{d\}\cup$ $\{$ forbidden assignment $\}$, and the latter from the fact that if there were two vertices from the same variable, there would be an edge between them. Note that in the last step is where we are using the fact that the maximum arity of the problem is greater than 1.

Remark. The previous remark also points out the fact that in general it's not true that the MC is a subgraph of the CPSG, since the former always contains edges of the form $x = 1, x = 2$ but the latter wouldn't have such edges if the arity of the problem is 1.

Proposition. π is an automorphism of the CPSG iff $\pi|_L$ is a CPS.

Proof. $\Rightarrow$ Let π be an automorphism of the CPSG. We have to prove that i) $\pi|_L$ is a solution symmetry and ii) if $a \in \cup_i \bar{C}_i$,$\pi|_L(a) \in \cup_i \bar{C}_i$.
i) By the first lemma, if s is a solution, $s \cup \{d\}$ is a set of $n+1$ quasi-independent vertices. Since $\{d\}$ is the only monary edge of the hypergraph, and π is an automorphism, $\pi(d) = d$. Therefore, $\pi(s \cup \{d\}) = \pi(s) \cup \{d\}$, and $\pi(s) \cup \{d\}$ is a quasi-independent set of $n+1$ vertices, since if there was an edge other than $\{d\}$ in $\pi(s) \cup \{d\}$, by applying the inverse automorphism π^{-1} to it, $s \cup \{d\}$ would have an edge which is not $\{d\}$, which is a contradiction.
ii) $a \in \cup_i \bar{C}_i \Rightarrow a$ is a PA forbidden by a constraint $\Rightarrow a \cup \{d\}$ is an edge of the CPSG. Since π is an automorphism, $\pi(a \cup d)$ is also an edge and we know that $\pi(d) = d$. The only edges to which d belongs are those of the form $\{d\} \cup \{$ PA forbidden by a constraint $\}$ and, therefore, we conclude that $\pi(a)$ is a forbidden PA, so $\pi(a) \in \cup_i \bar{C}_i$.

$\Leftarrow$ Let ϕ be a CPS. The only automorphism π of the hypergraph such that $\pi|_L = \phi$ is that which fixes d. We have to prove that π maps edges to edges.
a) Suppose s is a set of cardinality $m \leq k$ not forming a valid assignment. Suppose the image $\pi(s)$ is not an edge. ote that d is not in $\pi(s)$, because $\pi(d) = d$. A set of literals of cardinality lesser of equal than k can only be of two types. The first is a forbidden constraint; but, if $\pi(s)$ is a forbidden constraint, as $\pi|_L = \phi$ permutes them, $\pi^{-1}(s)$ must be a forbidden constraint, which is contradiction. The other type of set $\pi(s)$ can be is a nogood, so it can be extended to a solution of the problem: $\tilde{s}$. Now by the second lemma, if $\pi|_L$ is a CPS, $\pi^{-1}|_L$ is a CPS, so

$\pi^{-1}|_L(\tilde{s})$ is a solution too (because CPS are solution symmetries), but that would imply that s, which is a subset of $\pi^{-1}(\tilde{s})$, is a valid assignment, which is a contradiction.
b) Let s be a forbidden PA, so that $s \cup \{d\}$ is an edge. Since $s \in \cup_i \bar{C}_i$ and $\pi|_L$ is a constraint symmetry, we have $\pi|_L(s) \in \cup_i \bar{C}_i$. Therefore $\pi(s \cup \{d\}) = \pi(s) \cup \pi(d) = \pi|_L(s) \cup \{d\}$ is an edge.
c) Suppose s is an m-ary nogood, $m \leq k$, and assume $\pi(s)$ isn't an edge. This means $\pi(s)$ can be of two types, just as for the a) case, and the same proof applies
d) The automorphism π was chosen to fix d, so the $\{d\}$ edge maps to itself.

3.2 Microstructure Complement Definition

The Microstructure Complement (MC) of a CSP is a hypergraph whose vertex are the sets of all variable-value pairs (literals). A set of vertices $\{(v_1, a_1), (v_2, a_2), \ldots, (v_k, a_k)\}$ is an hyperedge of the MC iff:

a) $\{v_1, v_2 \ldots v_k\}$ is the set of variables in the scope of some constraint , but the constraint disallows the assignment $\{(v_1, a_1), (v_2, a_2), \ldots, (v_k, a_k)\}$; *or*

b) $k = 2,\ v_1 = v_2$ and $a_1 \neq a_2$.

Any automorphism of MC is a symmetry of the CSP. Constraint symmetries have been formally defined as the automorphisms of MC. It has been assumed implicitly that this definition is identical to the constraint preserving one detailed above. Let's have a closer look at the MC automorphisms and examine that assumption.

Claim. An automorphism of the MC can either be a constraint-preserving symmetry or a non-compositional symmetry, but not both at the same time.

Proof. It is easy to check that a symmetry is compositional iff every pair of literals of a same variable is mapped to a pair of literals of a same variable. Suppose there is a MC automorphism which is a non-compositional symmetry that preserves the constraints. If it preserves the constraints, every forbidden assignment edge is sent to another forbidden assignment edge, meaning that the edges associated to pairs of forbidden assignments are permuted between themselves by the symmetry. Thus, the preimage of a pair of literals of different variables is always a pair of literals of different variables. In consequence, the image of a pair of literals of a same variable is a pair of the same variable and the symmetry must be compositional.

Therefore an automorphism is either constraint preserving and compositional; or not constraint preserving and not compositional. From this result, two facts can be derived.

The set of MC automorphisms does not contain all constraint preserving symmetries.
Given the above claim, the validity of this affirmation is subject to the existence of non-compositional symmetries preserving the constraints, since they are out of the scope of MC. The existence of such symmetries is easily verified with an example:

$$CSP = \begin{cases} Variables\ \{x, y, z, t\} \\ Domain\ \{0, 1\} \\ Constraints \equiv \begin{cases} x + z + t = 4 \\ y + z + t = 4 \end{cases} \end{cases}$$

$$Symmetry\ (x = 0\ y = 1)$$

Thus MC never finds any symmetry in the left-down box of our symmetry subdivision table, which in fact contains the great majority of the potential constraint preserving symmetries.

The set of MC automorphisms contains symmetries which do not preserve the constraints This is easily verified with an example of automorphism of MC not preserving the constraints. Of course, according to the above claim, this is equivalent to the existence of a non-compositional automorphism of MC:

$$CSP \begin{cases} Variables\ \{x, y\} \\ Domain\ \{0, 1\} \\ Constraint\ x + y = 0 \\ Permutation\ (\text{x=0} \quad \text{y} = 0) \end{cases}$$
$$Automorphism\ (x = 0\ y = 0)$$

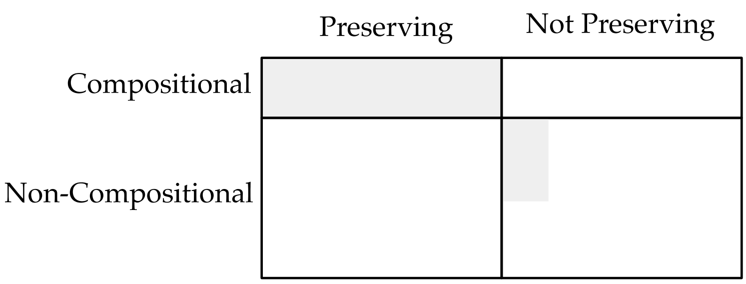

Figure 3: Synnetries found by MC.

According to the claim above, the automorphisms of MC can be of two types. A further interesting question is if MC finds all symmetries of both types. We begin by the compositional, constraint preserving symmetries. It is easy to see that all symmetries of this kind are automorphisms of the MC. Indeed, remember that a symmetry is compositional iff the image of every pair of literals of a same variable is a pair of literals of a same variable. Or equivalently, iff the image of every pair of literals of different variable maps to a pair of literals of different variables. This means that in any such symmetry the edges of type b) permute. And if the symmetry preserves the constraints, the edges of type a) permute also. Therefore any compositional CP symmetry is an automorphism of MC.

Instead, not all symmetries of the second type –those non-compositional and not preserving the constraints– are automorphisms of the MC. Here is an example:

$$CSP = \begin{cases} Variables\ \{x, y, z\} \\ Domain\ \{0, 1\} \\ Constraints \equiv \begin{cases} x + y = 0 \\ x + y + z = 0 \end{cases} \end{cases}$$
$$Symmetry\ (x = 0\ z = 0)$$

Indeed, one of the tuples disallowed by the first constraint is $\{(x, 0), (y, 1)\}$, which is mapped by the symmetry to $\{(z, 0), (y, 1)\}$. This tuple is nor disallowed by a constraint, neither belonging to the b) edges.

Finally, we examine in which cases, an automorphism of MC is constraint preserving (and compositional) or not (and is not compositional).

Claim. An automorphism π of the MC is a constraint-preserving symmetry iff π is an automorphism of the MC permuting the hyperedges of type b).

Proof. $\Rightarrow$ An automorphism of π permutes de edges of the MC. If π permutes also the a) edges. If π is also a CPS, it permutes the constraints and also the a) edges. Thus, the complement of this subset, the b) edges, must permute.
$\Leftarrow$ An automorphism of the MC is a symmetry and permutes all its edges. If besides, it permutes a subset, the b) hyperedges, it must permute the complement of the subset, i.e., the edges of type a) representing the constraints. Therefore π is a constraint preserving symmetry.

Since not permuting the b) edges is equivalent to having at least a b) edge mapped to a 2-constraint forbidden tuples (and viceversa), a direct consequence of this result is :

For problems without 2-constraints, MC automorphisms are always compositional symmetries.

Moreover, the number of 2-constraint disallowed tuples mapped to b) edges can be considered an index of maximal non-composionality. Therefore, even for CSPs with 2-ary constraints, the possibilities of non-compositional symmetries are limited (among other things) by the cardinality of those constraints.

3.3 Relation between the two definitions

A conclusion of the two preceding sections is that the intersection of the two possible definitions of CS is the set of compositional constraint-preserving symmetries. Why?

The first definition is theoretical. The second one is a constructive definition and closer to practice; it can be considered as an attempt to implement the theoretical definition. Any method trying to find all CPSs is conditioned by the result in the previous section: a constraint preserving permutation of literals must map solutions to points to be a symmetry. But without the knowledge of the solutions, how any method could determine which constraint-preserving literal permutations are symmetries? The only way of not risking to output non-symmetries is to only select permutations that map *every* point to a point. But it can be shown that any symmetry that maps points to points is a compositional symmetry. Therefore, it is natural that the only constraint preserving symmetries found by MC are compositional (and that the non-compositional symmetries it finds do not preserve the constraints).

Composionality brings closer the concepts of constraint preservation and symmetry, as the following lemma points out:

Lemma. Given a compositional permutation s, if s preserves the constraints, then s is a symmetry.

Proof. This is a direct consequence of the alternative definition of Constraint Symmetries not requiring the permutation to be a symmetry in the first place. Since s is compositional, it maps points to points, and in particular, solutions to points. Thus, it matches the alternative CPS definition.

The converse is not true, as there are compositional solution symmetries that do not preserve the constraints like the following one:

$$CSP = \begin{cases} Variables\ \{x, y, z\} \\ Domain\ \{0, 1\} \\ Constraint \equiv \begin{cases} x + y = 0 \\ x + y + z = 0 \end{cases} \end{cases}$$
$$Symmetry\ (x = 0\ z = 0)(x = 1\ z = 1)$$

However, given a compositional permutation, there exists a condition that is equivalent to being a symmetry: preserving the set of extended constraints, as the following proposition claims. Let the extended constraint $\bar{C}_i^*$ denote the set of points disallowed by the i-th constraint.

Lemma. Let s be a compositional permutation. Then s is a (solution) symmetry of $P =< X, D, C >$ if and only if $s(\cup_i \bar{C}_i^*) = \cup_i \bar{C}_i^*$.

Proof. It is easy to check that the problems P and $P^* =< X, D, C^* >$ have the same solutions (P^* is a reformulation of P) and that the solutions of P^* (and thus, of P) are $\{x\ s.t.\ x \notin \cup_i \bar{C}_i^*\}$
$\Rightarrow$ Assume s is a symmetry of P and there exists a point $p \in \cup_i \bar{C}_i^*$ such that $s(p) \notin \cup_i \bar{C}_i^*$. Since s is compositional, $s(p)$ must be a point. Therefore, p is not a solution and $s(p)$. is a solution. Since s is a symmetry, also s^{-1} is a symmetry, but s^{-1} maps a solution, $s(p)$, to a non-solution, p, and, thus, does not preserve solutions, which is a contradiction.
$\Leftarrow$ A permutation s of L induces a permutation on its powerset, $\mathcal{P}(L)$. If s is compositional, we know that points are mapped to points. The set of points is a subset of $\mathcal{P}(L)$. Therefore, the restriction of the induced $\mathcal{P}(L)$ permutation to the set of points is a permutation of the points of the domain. Thus, on the one hand s permutes the points of the domain. On the other hand, by hypothesis s permutes the points in $\cup_i \bar{C}_i^*$, which is the set of non solutions. Thus, its complement in the set of points, the set of solutions, also permutes. Thus, s preserves the solutions and is a symmetry.

This means that the whole first row in our tables preserves the extended constraints.

Among CP symmetries, compositional ones are interesting also because of another reason. As said before, compositional symmetries map points of the domain to points of the domain. There are very well established methods available to deal with this kind of symmetries (as Lex-Leader) not applicable in the general case.

Thus it would be interesting to have a graph finding all the compositional CP symmetries and only them. A modified version of the MC graph, which we call MC', can be used for this purpose. A dummy vertex, d must be added to the MC. A new hyperedge containing only d is also added. And the binary hyperedges of type b) of MC, become ternary hyperedges by including in all of them the new vertex d.

Proposition. π is an automorphism of MC' iff $\pi|_L$ is a compositional CPS

Proof. $\Rightarrow$ Note that, since $\{d\}$ is the only unitary hyperedge, d must map always to d in an automorphism of MC', and thus the remaining vertices (the literals), can only be mapped to themselves. Therefore, an automorphism of MC' is a bijection of the literals.

Let an enlarged point be the set of literals of a point jointly with d. Thus, an enlarged point does not contain any of the d-ternary edges. An automorphism of MC' permutes these edges because they are the only ternary edges containing d and d maps to itself. Therefore, there are no d-ternary edges in the image of an enlarged point. The subset of n literals in the enlarged point should map to n literals because d maps to itself. As there are not d-ternary edges in the enlarged point, there are not two literals of the same variable in its image. Thus, points are mapped to points.

In consequence an enlarged solution point does not contain neither d-ternary nor a) edges (or it would not be a solution): it does only contain the monary hyperedge $\{d\}$. Because of this, the image of an enlarged solution in a MC' automorphism contains also only a monary edge, which can only be $\{d\}$. Since the set of literals of the image is a point (because it is the image of a point) and does not contain an a) edge, it is a solution. We have shown that an automorphism of MC' maps points to points and solutions to solutions. Thus, only compositional symmetries are automorphisms of this graph. Finally, it is easy to see that these symmetries must be constraint preserving. Indeed, since d maps to d and the enlarged b) edges permute, the remaining edges, i.e., the a) edges representing the constraints are also permuted, for which the constraint are preserved.
$\Leftarrow$ Let π a permutation of $L \cup \{d\}$ such that $\pi|_L$ is a compositional CP symmetry and $\pi(d) = d$. We have to show that every edge of the MC' is sent by $\pi|_L$ to another edge of the MC'.
i) d-ternary edges (i.e, those of the form $\{d\} \cup \{$ vertex of type b) in the MC $\}$. Since $\pi|_L$ is compositional, it sends vertices of type b) to vertices of type b) in MC. And, since $\pi(d) = d$, d-ternary edges are sent to d-ternary edges in MC'.
ii) $\{d\}$. As stated before, d is a fixed point by π.
iii) Forbidden assignments. This follows directly from the following property of the CCS: if $s \in \cup_i \bar{C}_i$, $\pi|_L(s) \in \cup_i \bar{C}_i$.

Although a new vertex has been included, the search for automorphisms is less hard in MC', because, having the same number of hyperedges (well, only one more that can only be mapped to itself), the modified b) hyperedges cannot be exchanged with those corresponding to 2-constraints.

Interestingly, this graph can also be used to go beyond the intersection of the two definitions, the group of compositional CP symmetries. Indeed, the group of automorphisms of the MC' of problem $P^* = (V, D, C^*)$ is the group of compositional symmetries (*both constraint preserving and not*) of problem $P = (V, D, C)$. This is illustrated in Figure 4, and proved in the proposition below. Note that the last example showed that this graph finds indeed something new, that is, that the upper-right box of the table is not empty.

	Preserving	Not Preserving
Compositional		
Non-Compositional		

Figure 4: Symmetries of problem $P = (V, D, C)$ identified by the automorphisms of the MC' of problem $P^* = (V, D, C^*)$.

Proposition. π is an automorphism of the MC' of $P^* =< X, D, C^* >$ iff $\pi|_L$ is a compositional symmetry of $P =< X, D, C >$

Proof. Using the last lemma and renaming $\pi|_L$ as ϕ, we can restate the proposition like this:

ϕ is is a compositional CPS of $P^* =< X, D, C^* >$ iff ϕ is a compositional symmetry of $P =< X, D, C >$.

$\Rightarrow$ Taking into account that P^* is a reformulation of P and, therefore, both problems have the

same symmetries, this implication is obvious.
$\Leftarrow$ Let ϕ be a compositional symmetry of P. A previous lemma says that if ϕ is a compositional symmetry of P then ϕ is a permutation preserving C^*. Since a symmetry of P is also a symmetry of P^* and a compositional symmetry of P is also a compositional symmetry of P^* we have that ϕ is a compositional symmetry of P^* preserving its constraints.

Thus, we can identify all compositional symmetries, even those not preserving the constraints, without the need of explicit solution information. This was not the case for identifying all CP symmetries (see Section 3.1) and known graphs not relying on nogoods, like MC, cannot find any non-compositional CP symmetry. Conclusion: it is not to be constraint preserving the feature which makes a symmetry to be easily identifiable, but to be compositional.

Compositional symmetries have another important property. Compositionality means that the literals of a given variable can only be mapped to literals of a unique variable. And as consequence, the assignments of a constraint map eventually to assignments of different constraints, *but having all of them the same scope.* For problems in which each constraint has a different scope, each constraint as a whole maps to a different constraint and, in fact, the whole constraints permute. That is, if s is compositional then $s(C_i) = S(C_{\psi(i)})$, where ψ is a permutation of the indices of the constraints. But this adds a new restriction: only constraints with the same cardinality can permute. For CSPs having several constraints with the same scope, the situation is the same considering each group of constraints with the same scope as a super-constraint. The groups permute and each group can only map to a group wih a sa cardinality of the union of the forbidden tuples in the group[1].

In the CP definition of Constraint Symmetries there exist the freedom to map arbitrarily a tuple of a constraint to a tuple of any constraint, since only the set of tuples of all constraints must be preserved. However, in the CP automorphisms of MC this freedom is superficial. All CP symmetries found by MC exchanges constraints by blocks, in the sense commented above, because they must be by force compositional.

3.4 Extending MC to find other non-preserving symmetries

The automorphisms of MC potentially includes non-constraint-preserving symmetries when the CSP includes constraints of arity 2. Thus, the identification of these symmetries does not require explicit knowledge of the set of solutions, since neither the creation of the MC graph requires it.

It is possible to create a hypergraph with the same logic of MC (and also without explicit knowledge of the solutions) which can include non-CPS symmetries no matter the arity of the CSP. Consider the hypergraph associated to a k-ary CSP whose vertices are the literals and having the following sets of hyperedges:
a) Sets of cardinality m for $m \leq k$ not forming a valid PA .
b) Each PA explicitly forbidden by a constraint, that is, any PA a such that $a \in \cup_i \bar{C}_i$

The MC graph is a subgraph of this graph which at the same time is a subgraph of KNG. Every automorphism of this graph is a symmetry (A set of vertices is a solution iff it is an independent set of size n and any automorphism must map independent sets to independent sets). Just like in MC, it can be easily shown that if an automorphism of the extended MC is constraint-preserving, then it's compositional, and if it's compositional, it's constraint-preserving. And all compositional CPSs are automorphisms of the extended MC as well.

For the special case of problems of arity greater than two and containing constraints of arity two, there is not a clear inclusion relation; the extended MC finds also symmetries which are not MC automorphisms (as the example below shows), but the opposite is in principle also possible.

[1] the tuples forbidden by several constraints must be counted only once

Therefore, the automorphisms of the MC are not always completely included in those of the extended MC. In turn, this implies that the extended MC does not either fill the down-right box. All this is reflected in Figure 5.

In despite of this, the automorphisms of the MC are always completely included for all problems outside the special case mentioned above.

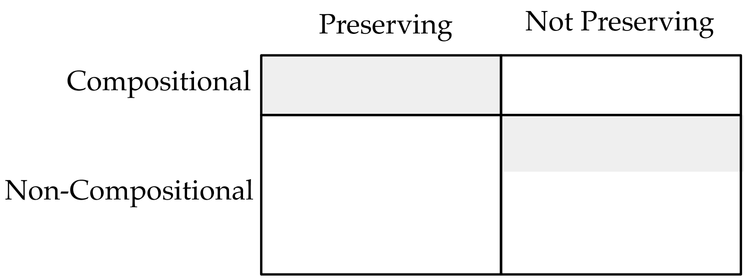

Figure 5: Synnetries found by the extended MC.

This is an example of symmetry in a problem with constraints of arity two and higher in the down-right box of the figure found by ExtendedMC and not found by MC:

$$CSP = \begin{cases} Variables\ \{x, y, z\} \\ Domain\ \{0, 1\} \\ Constraints \equiv \begin{cases} x + y + z = 0 \\ x + z \neq 1 \end{cases} \end{cases}$$
$$Symmetry\ (x = 0\ z = 0)$$

The symmetry is not compositional and does not preserve the constraints. It is an automorphism of the MC since the tuple forbidden by the first constraint $\{(x, 0), (y, 1), (z, 1)\}$ is mapped to an invalid point and thus, not an edge. Instead, any tuple of cardinality three is an a) or b) edge of ExtendedMC (except the solution, $\{(x, 0), (y, 0), (z, 0)\}$, which maps to itself) and therefore must permute between themselves. It is easy to check that the pairs of literals forbidden by the second constraint jointly with $\{(x, 0), (x, 1)\}$ and $\{(z, 0), (z, 1)\}$ permutes between themselves and that the other invalid pair, $\{(y, 0), (y, 1)\}$, maps to itself, for which the symmetry is an extended MC automorphism.

An example of symmetry in the down-right box not found by ExtendedMC is the one used in Section 3.1.2 for the same purpose with MC.

4 Constraint Symmetries in the context of point symmetry

A constraint symmetry s is a point symmetry such that $s(\cup_i \bar{C}_i^*) = \cup_i \bar{C}_i^*$.

This definition has unexpected implications. A permutation of a set that permutes a subset also permutes the complement of the subset and viceversa. Therefore, a constraint symmetry is a symmetry s such that $s((\cup_i \bar{C}_i^*)^c) = (\cup_i \bar{C}_i^*)^c$. But $(\cup_i \bar{C}_i^*)^c = \bigcap_i C_i^*$, which is the set of points allowed by all constraints. That is, the set of solution symmetries. Therefore, a general constraint symmetry is completely equivalent to a solution symmetry.

Thus, both for literal symmetries and for point symmetries, the concept of general constraint symmetries is not useful. In the first case, because they are not easy to detect and in the second case because it is redundant concept. Also, as in the context of literal symmetries it may be useful to consider subsets of point constraint symmetries (which coincide with solution symmetries), like compositional or those exchanging whole constraints or even subsets of constraints.

5 Conclusions

The property of preserving constraints is not valuable by itself in a symmetry. It is supposed to be key for a literal symmetry to be detected easily. But, first, we have shown that it is equally easy to detect symmetries not preserving constraints, and in fact, even the MC is able to do it. And, second to identify most of the symmetries preserving the constraints requires the knowledge of the no-goods of the problem (and, in fact, they are not found by MC). For point symmetries (permutations of domain points) Constraint Symmetries result to be superfluous, because they are the same set as the solution symmetries. Thus, in conclusion, the concept of general constraint symmetries is irrelevant for practical symmetry detection. However distinguished subsets of them *are* relevant. For example, the subgroup of compositional constraint-preserving symmetries is easily found without knowledge of the solutions and is in fact completely included in the automorphisms of the MC. Actually, these are the only symmetries preserving the constraints found by the MC.

This is natural in the light of the detailed examination of the constraint-preservation made in this paper. The constraint-preserving property does not confer the status of symmetry to a literal permutation: mapping every solution to a point is necessary and sufficient condition for conferring such status. But, without solving previously the problem, the only way to guarantee that a constraint-preserving permutation is a symmetry is requiring that every point maps to a point. And this requirement results to be equivalent to compositionality. Therefore, the only constraint-preserving symmetries that any method can find are compositional ones (at least if only symmetries are wanted as output). We can arrive to a reciprocal conclusion for compositional symmetries. For a compositional permutation, preserving the constraints is an enough condition to be a symmetry. However, not all compositional symmetries are constraint preserving; Instead, preserving the extended (negative) constraints is a necessry and sufficient condition, but this is equivalent to preserving the solutions. Therefore, the knowledge of the solution set is required to find all compositional symmetries and only the constraint-preserving ones are identifieable without that knowledge.

A variant of MC finding solely these symmetries can be designed. In the opposite direction, a variant of MC can be designed that in addition to the compositional preserving symmetries is able to find more non-constraint-preserving symmetries than plain MC without the knowledge of the no-goods of the problem. Finally another distinguished subgroup of Constraint symmetries seems that of Block Constraint symmetries, in which whole constraints or set of constraints are exchanged. We have found out some of their relations with compositional symmetries, but this is worth of further study as well as how they relate to identifiability.

References

[1] K. Apt, “Principles of constraint programming,” Cambridge University Press, 2003.

[2] D. A. Cohen, P. Jeavons, C. Jefferson, K. E. Petrie, B. M. Smith, “Symmetry Definitions for Constraint Satisfaction Problems,” in *van Beek P. (eds) Principles and Practice of Constraint Programming - CP 2005*, CP2005, Lecture Notes in Computer Science, vol 3709, pp. 17-31, Springer, 2005.

[3] COPRIN, The COPRIN examples page, http://www-sop.inria.fr/coprin/logiciels/ALIAS/Benches/, 2011.

[4] I. P. Gent, K. E. Petrie, J.-F Puget, "Symmetry in Constraint Programming", in Handbook of Constraint Programming, pp. 329-376, 2006.

[5] B. D. McKay and A. Piperno, "Practical Graph Isomorphism, II," Journal of Symbolic Computation, vol. 60, pp. 94-112, 2014.

[6] J.-F. Puget, "Automatic Detection of Variable and Value Symmetries," in *van Beek P. (eds) Principles and Practice of Constraint Programming - CP 2005*, CP2005, Lecture Notes in Computer Science, vol. 3709, pp. 475-489, 2005.

[7] V. Ruiz de Angulo and C. Torras. "Exploiting single-cycle symmetries in continuous constraint problems,". Journal of Artificial Intelligence Research, vol. 34, pp. 499-520, 2009.

[8] G. Alexandre, J. Cristophe, V. Ruiz de Angulo and C. Torras: "Variable symmetry breaking in numerical constraint problems." Artificial Intelligence, vol 229, pp. 105-125, 2015.

[9] C. Mears, M. Garcia de la Banda, M. Wallace. "On Implementing Symmetry Detection," Constraints, vol. 14, pp. 443-477, Springer, 2009.

Acknowledgements

This work was supported by IRI-CFIS Internship Programme, supported by the Spanish State Research Agency through the Maria de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

IRI reports

This report is in the series of IRI technical reports.
All IRI technical reports are available for download at the IRI website
http://www.iri.upc.edu.