

Foundations of Symmetry Breaking Revisited–Revised

Tim Januschowski,^{*} Barbara M. Smith,[†] and M.R.C. van Dongen[‡]
janus@cs.ucc.ie

Abstract

Puget’s article “On the Satisfiability of Symmetrical Constraint Satisfaction Problems” [Puget, 1993] is widely acknowledged to be one of the founding papers in symmetry breaking (see for example [Flener *et al.*, 2006; Rossi *et al.*, 2006; Walsh, 2006]). To date, Puget’s definition of a valid reduction seems to be the only common ground that all authors on static symmetry breaking constraints agree on. His original definition of a valid reduction is for a restricted form of variable symmetries. We extend Puget’s definition to the recent, more general definition of symmetries by Cohen *et al.* [Cohen *et al.*, 2006]. In our extension, we require a valid reduction to be a tighter constraint satisfaction problem instead of one with more constraints. We re-prove both his central theorems on valid reductions for our definition and we prove considerably stronger results on the existence of valid reductions with further properties. Along the way, we present examples which contradict the common belief that we can always eliminate all symmetries with lexicographic ordering constraints.

1 Introduction

Symmetries are a common property of Constraint Satisfaction Problems (CSPs). In practice, symmetries often are a key to an effective solution of symmetric CSPs: *only if we properly exploit the symmetries may we solve such CSPs within a reasonable time.* The classic approach to dealing with symmetries is the addition of dedicated constraints before search. These constraints are traditionally called *symmetry breaking constraints*. This is the most commonly used symmetry breaking approach in practice and it receives consid-

erable attention. We consider the theoretical foundations of the addition of symmetry breaking constraints in the present work following Puget [Puget, 1993].

Puget’s approach to introduce symmetry breaking constraints [Puget, 1993] for a given CSP is via the definition of a related CSP that he calls a *valid reduction*. Although he presents a specific symmetry breaking constraint to prove the existence of a valid reduction, valid reductions themselves are defined without dependence on any specific symmetry breaking constraint. To date, valid reductions seem to be the only common ground that authors on symmetry breaking constraints agree on. The definition of a valid reduction is the foundation of all symmetry breaking constraints as diverse as they may be: general lexicographic ordering constraints [Crawford *et al.*, 1996], column-inequalities in Integer Programming [Kaibel and Pfetsch, 2008] or problem-specific symmetry breaking constraints in Satisfiability [Shlyakhter, 2007] and in Constraint Programming (CP) [Cambazard *et al.*, 2009]. These constraints differ in so many aspects, that valid reductions may offer the only framework to compare the different symmetry breaking constraints. Furthermore, valid reductions offer a means to study the theoretical effects of the addition of symmetry breaking constraints. This is the reason why an investigation of valid reductions is important for the current research on symmetry breaking constraints. We see valid reductions as a first step towards a general theory of symmetry breaking constraints.

In the present work, we informally revisit Puget’s definition of a valid reduction. Puget crafted his definition for a restricted form of variable symmetry. However, the recent definition of symmetries by Cohen *et al.* [Cohen *et al.*, 2006] is much wider. We provide their formal definition in Section 2 where we also define our notation. Cohen *et al.* identify the microstructure complement as the basic structure for reasoning about symmetries. In Section 3.1 we carry Puget’s definition of a valid reduction over to the recent, more general definition of symmetry by Cohen *et al.* [Cohen *et al.*, 2006] using microstructure complements. We add a slightly different emphasis to the original definition by requiring a valid reduction to be a tighter CSP than the original CSP. Our focus on microstructure complements makes this shift in emphasis seem natural. With a simple example in Section 3.2, we show that adding Puget’s constraints may not change the CSP. This underlines the importance of our change of em-

^{*}Tim Januschowski is supported by the EMBARK initiative by the Irish Research Council for Science, Engineering and Technology; Cork Constraint Computation Centre and Computer Science Department, University College Cork, Ireland
janus@cs.ucc.ie

[†]School of Computing, University of Leeds, UK

[‡]Computer Science Department, University College Cork, Ireland

phasis in the definition. The example shows a trivial alternative to prove Puget’s main existence theorem. Therefore, we suggest that Puget’s constraint with which he proves the existence of a valid reduction is his main contribution. With another example, we question the usual interpretation of his paper, namely that “Puget proved that symmetries can always be eliminated by the addition of suitable constraints” (Walsh [Walsh, 2006]). We present a simple example of a valid reduction that may still have some of the original symmetries when we add the ordering constraints that Puget suggests. We identify the cases in which we can find a valid reduction that differs from the original CSP and we prove stronger versions of Puget’s central theorems. We show an example of a CSP that has symmetries, which cannot be eliminated with the traditional methods. To the best of our knowledge, it thus really is an open question when we can eliminate symmetries. In Section 3.3, we consider valid reductions with further properties. We summarise our work in Section 4. We present open questions along the way.

2 Preliminary Definitions

Let us introduce some notation and graph theoretic concepts that we will need in the following. We define a CSP P as the triple (X, D, C) , where X is the list of variables of P , $D(x)$ is the domain of the variable $x \in X$, and where C is the set of constraints of P . Since we are interested in theoretical results, we mainly consider constraints in extensional form. Each constraint c has an arity k and consists of a scope and a relation. The scope of a k -ary constraint c is a sequence $\langle x_1, \dots, x_k \rangle$ and the relation of c is a subset of the Cartesian product of the domains of the variables in the scope. We say that the tuples in the relation of c are *allowed* by c . All other tuples are *forbidden* by c . We also call a forbidden tuple a *no-good*. For ease of presentation, we exclude CSPs that have variables with empty domains. We further assume that in the domain of a variable no values exist that are forbidden by a unary constraint.

We call a (variable,value)-assignment a *literal*. A solution of P is an assignment of values to all variables that is allowed by all constraints. If a solution exists, we say that P is satisfiable, otherwise P is unsatisfiable. The following is an example of a CSP.

Example 1. Consider a CSP with 2 variables x and y , each with domain $\{1, 2\}$ and one extensional constraint that only allows $\{(x, 2), (y, 2)\}$ and forbids all other tuples. This CSP is satisfiable and the only solution is $\{(x, 2), (y, 2)\}$.

We typically denote a hypergraph G as a tuple (V, E) with V or $V(G)$ the set of nodes and E or $E(G) \subseteq 2^V$ the set of hyperedges. For $\hat{V} \subseteq V$, we denote the *induced subgraphs* of G by $G[\hat{V}]$. The induced subgraph is a hypergraph $G[\hat{V}] = (\hat{V}, \hat{E})$, where $\hat{E} = \{ \{v_1, \dots, v_r\} \in E : v_1, \dots, v_r \in \hat{V} \}$, i.e., all nodes of every edge in $G[\hat{V}]$ are in \hat{V} . Let us associate a hypergraph to every CSP instance.

Definition 2 (Microstructure Complement [Freuder, 1991; Jégou, 1993]). *With every CSP $P = (X, D, C)$ we associate*

a hypergraph G . We call this hypergraph the microstructure complement (MSC) of P . The nodes of G are the elements of the set of literals $\bigcup_{x \in X} \{x\} \times D(x)$. A set of nodes $\{(x_1, t_1), (x_2, t_2), \dots, (x_k, t_k)\}$ is a hyperedge of the MSC if:

- *a constraint in P with scope $\{x_1, x_2, \dots, x_k\}, k > 1$ exists that forbids the tuple $\langle t_1, \dots, t_k \rangle$*
- *$k = 2, x_1 = x_2$ and $t_1 \neq t_2$.*

For binary CSPs, this graph is the complement of the *microstructure* [Freuder, 1991; Jégou, 1993], as the name microstructure complement suggests. We want to define the MSC as a loop-free hypergraph, which explains why $k > 1$. More specifically, we assume that literals forbidden by unary constraints do not appear as nodes in the MSC. The MSC is a simple hypergraph, i.e., it does not have parallel hyperedges.

A *stable set* in a hypergraph G is a set of nodes, \hat{V} , such that $G[\hat{V}]$ does not contain any hyperedge. The MSC that corresponds to some CSP with n variables admits a maximum stable set of size at most n . A CSP with n variables is satisfiable if and only if its MSC contains a stable set with n nodes. Each stable set with n nodes corresponds to a solution in the CSP and vice versa. A *clique* \hat{V} is a set of nodes in the graph G such that $G[\hat{V}]$ is the complete graph. In a binary CSP P with n variables, each clique with n nodes in the microstructure of P corresponds to a solution and vice versa.

In Figure 1a, we have depicted the MSC of the CSP in Example 1. The MSC contains one maximum stable set of size 2 consisting of the nodes $\{(x, 2), (y, 2)\}$. The maximum stable set corresponds to the only solution of the CSP.

We remind the reader of the following graph theoretic definition that is the basis for our definition of symmetries for CSPs.

Definition 3 (Graph Automorphism). *Let $G = (V, E)$ be a hypergraph and let $\phi : V \rightarrow V$ be a bijective function. For $\hat{V} \subseteq V$, we shall write $\phi(\hat{V})$ to denote $\bigcup_{v \in \hat{V}} \{\phi(v)\}$. We call ϕ a graph automorphism if it has the following property: for any hyperedge e of G , $\phi(e)$ also is a hyperedge of G .*

Note, that any automorphism also maps non-hyperedges to non-hyperedges. We denote the group of automorphisms of a hypergraph G by $aut(G)$. The *action* of the group $aut(G)$ on the set of nodes is the permutation of the set of nodes as a result of the automorphisms. We say that $aut(G)$ *acts on* the set of nodes. The *orbit* of a set of nodes \hat{V} in G is the set of all images of \hat{V} under the action of $aut(G)$. We denote it by $orbit(\hat{V}, aut(G))$ or simply $orbit(\hat{V})$ if it is clear which group is meant. From now on, we shall write graph instead of hypergraph. The following definition, by Cohen *et al.* [Cohen *et al.*, 2006], is the current state-of-the-art for symmetry definitions in CP.

Definition 4 (Constraint Symmetry [Cohen *et al.*, 2006]). *Let G be the MSC of a CSP P . A constraint symmetry of P is an element of $aut(G)$.*

We shall abbreviate constraint symmetries to symmetries. Cohen *et al.* [Cohen *et al.*, 2006] show that another commonly used definition for symmetries in CP is equivalent to Definition 4 with an enhanced MSC. They call this symmetry *solution symmetry*. Our results can be carried over for solution

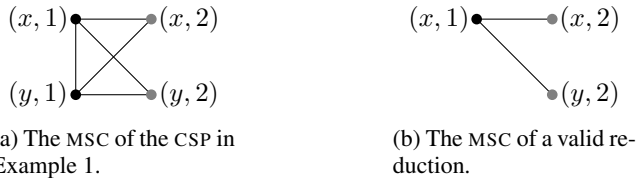


Figure 1: The MSC of the CSP in Example 1 and one of its valid reductions. The solution $\{(x, 2), (y, 2)\}$ is marked in gray. The solution is invariant under the symmetries and is a maximum stable set.

symmetries. However, due to space restrictions, we shall omit references to solution symmetries.

Symmetries act on the literals or the set of nodes in the MSC, i.e., for a CSP (X, D, C) , the symmetries act on $\bigcup_{x \in X} \{x\} \times D(x)$. We define a *variable symmetry* as a special class of symmetries. Symmetry ϕ is a *variable symmetry* if ϕ maps (x, v) to $(\psi(x), v)$, where ψ is some suitably chosen bijective function on the variables. We say that a variable symmetry *acts on the variables* of a CSP. In his paper, Puget [Puget, 1993] considers a restricted form of variable symmetry. He only allows variable symmetries that act on variables in the scope of a symmetric constraint.

In Example 1, the following symmetries are among the symmetries of the MSC: the identity, which is a symmetry of any CSP, and a self-inverse symmetry ϕ that swaps $(x, 1)$ with $(y, 1)$ and $(x, 2)$ with $(y, 2)$. The symmetry ϕ is a variable symmetry also in Puget’s sense. Definition 4 incorporates Puget’s definition of symmetries and allows for more symmetries. We refer the interested reader to [Cambazard *et al.*, 2009] for a particularly nice example of a symmetry which is not a variable symmetry.

Having defined what the symmetries of a CSP are, we proceed by defining a symmetric CSP. Clearly, the identity is an automorphism of any CSP. It is natural to exclude CSPs whose only automorphism is the identity from being called symmetric CSPs. We exclude another, arguably trivial, class of CSPs from being called symmetric. If every variable has not more than one value in its domain, satisfiability can be checked in polynomial time. If the CSP is satisfiable, the symmetry group consists of all permutations of the literals. Excluding this class of CSPs from being called symmetric, we obtain the following definition.

Definition 5 (Symmetric CSP). *A CSP having at least one variable with domain size strictly larger than 1 is symmetric if the size of the symmetry group is larger than 1.*

Note that an empty CSP is not symmetric according to this definition. As we have seen before, the CSP from Example 1 has one solution and admits several symmetries. Therefore, the CSP is symmetric and each symmetry maps this solution to itself. We now have all basic ingredients to extend Puget’s definition of a valid reduction in the following section.

3 Valid Reductions

Symmetries partition the set of solutions into orbits. Symmetry breaking constraints have the property that they allow

at least one solution in every orbit of solutions. Put differently, the addition of a symmetry breaking constraint should always produce a CSP with a representative of each orbit of solutions: an *orbit representative*. This is the defining property of Puget’s valid reduction.

3.1 Definition of Valid Reduction

Puget defines a *valid reduction* as a CSP with strictly more constraints than the original, possibly symmetric CSP such that for every orbit of solutions at least one orbit representative remains. He defines valid reductions for a restricted form of variable symmetries. We extend this to more general symmetries as in Definition 4. By focussing on the MSC as in Definition 4, it seems natural to substitute *fewer partial solutions* for *more constraints*. The following is our definition.

Definition 6 (Valid Reduction). *Let $P = (X, D, C)$ be a CSP with n variables, MSC G and symmetry group $\text{aut}(G)$. Let $\tilde{P} = (X, \tilde{D}, \tilde{C})$ be a CSP with the same variables as P , $\emptyset \neq \tilde{D}(x) \subseteq D(x)$ for all $x \in X$.*

We call \tilde{P} a valid reduction of P if the MSC \tilde{G} of \tilde{P} fulfils the following conditions:

1. *a) $V(\tilde{G}) \subseteq V(G)$ and b) $E(G[V(\tilde{G})]) \subseteq E(\tilde{G})$, and*
2. *for every maximum stable set S of size n in G , at least one element in $\text{orbit}(S)$ exists in \tilde{G} .*

Let us comment on this definition. According to the definition, we can produce a valid reduction in two ways only. The first way to produce a valid reduction is by removing nodes from the MSC, Condition 1a). This is equivalent to adding a unary constraint. The second way to produce a valid reduction is by adding hyperedges to the MSC of the CSP, Condition 1b). This is equivalent to adding a constraint of arity 2 or higher to the CSP. For microstructures of binary CSPs, we may produce a valid reduction by removing a node or removing an edge. Any constraint that we add to the original CSP to obtain a valid reduction cannot exclude entire orbits of solutions. The existence of an orbit representative ensures that we can reconstruct all solutions of the original CSP by applying the symmetries to the orbit representatives in the original CSP.

A valid reduction is defined in terms of removing symmetric solutions rather than eliminating symmetries. However, by removing symmetric solutions, we may also eliminate some symmetries.

Definition 6 does not state that a valid reduction is defined only for symmetric CSPs in the sense of Definition 5. We only refer to the group of automorphisms. This group always contains the identity. In this, we have followed Puget.

Definition 6 is an extension of Puget’s original definition since we allow for more general symmetries to define the orbits. We stress that we may add unary constraints to a CSP by removing nodes. Puget implicitly allows unary constraints in his definition, though “at the core of [his] method” is “adding ordering constraints” [Puget, 1993]. We disallow unary constraints that remove all literals corresponding to a variable. Though this may reduce symmetries, it may make it too difficult to relate a valid reduction to the original CSP and to reconstruct solutions. Further work will address this.

Definition 6 substitutes fewer stable sets in the MSC — they correspond to fewer partial or full solutions — for higher number of constraints. We ensure fewer stable sets in our definition by requiring the MSC of a valid reduction to either have fewer nodes (Condition 1a) or having more edges on the set of nodes in the MSC that the valid reduction and the original CSP have in common (Condition 1b). We shall say that a valid reduction with fewer stable sets in its MSC than the original MSC is *tighter*. This change in the definition is motivated by the fact that a higher number of constraints in Puget’s sense could mean a mere repetition of an already present constraint.

We defer a discussion on the existence of valid reductions to Section 3.2. First, we present a result that ties the satisfiability of a symmetric CSP to the satisfiability of a valid reduction of the CSP. Puget first proved this theorem. We extend it to the larger class of symmetries as in Definition 4 using our definition of valid reduction. Puget’s original theorem is a sufficient condition for the satisfiability of a symmetric CSP in terms of valid reductions. We strengthen it to provide a characterisation.

Theorem 7. *A symmetric CSP P is satisfiable if and only if any valid reduction of P is satisfiable.*

Proof. If the CSP P is satisfiable, then any valid reduction needs to have at least one solution by definition and is hence satisfiable.

If the CSP P is unsatisfiable, then its MSC does not have a stable set of size n , where n is the number of variables. According to Definition 6, the MSC of any valid reduction contains fewer nodes and possibly more edges when comparing induced subgraphs corresponding to the same set of nodes. Hence, the MSC of any valid reduction has a maximum stable set that is at most as large as the maximum stable set in the original MSC. \square

Since we did not use symmetries that are not the identity in the above proof, Theorem 7 holds for CSPs that are not symmetric as well.

Corollary 8. *A CSP P is satisfiable if and only if any valid reduction of P is satisfiable.*

3.2 Existence Of Valid Reductions

Besides Theorem 7, Puget’s other central result is the existence of a valid reduction for symmetric CSPs. Before we prove this result for our Definition 6, let us briefly discuss Puget’s proof for variable symmetries.

In a CSP with a variable symmetry ϕ that maps a variable x to a different variable y , Puget shows that the addition of the constraint $x \leq y$ to the original CSP produces a valid reduction. Indeed, the constraint does not exclude an entire orbit of solutions and this valid reduction has one constraint more than the original CSP.

We believe that inventing the symmetry breaking constraint $x \leq y$ is the more important achievement compared to the proof of the existence of a valid reduction. Let us clarify this using Example 1 where we have a symmetry between the variables x and y . This would have been a symmetry in Puget’s terminology as well. Since we add the constraint $x \leq y$ to the CSP, this valid reduction has a higher number of

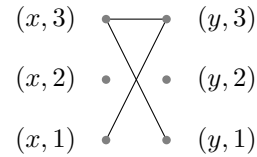


Figure 2: The microstructure of Example 9.

constraints. The addition of constraint $x \leq y$, however, does not change the MSC at all. Example 1 shows that producing a tighter valid reduction with Puget’s constraints is not always possible. If the existence theorem, rather than the symmetry breaking constraint, was the main result, then Puget could have proved the theorem also by introducing a constraint that is a copy of an already existing constraint.

Let us give another example that shows that adding ordering constraints to a symmetric CSP may result in valid reductions which are *more* symmetric than the original CSP, as promised in Section 1.

Example 9. *Consider a CSP with two variables x and y . The domain of both variables is $\{1, 2, 3\}$ and the constraint between x and y allows $\{(x, 1), (y, 3)\}$, $\{(x, 3), (y, 1)\}$ and $\{(x, 3), (y, 3)\}$. The microstructure of this CSP is depicted in Figure 2. Let us consider two symmetries in particular. The first symmetry, ψ , swaps $(x, 2)$ with $(y, 2)$ and is the identity on all other literals. The second symmetry, ϕ , swaps $(x, 3)$ with $(y, 3)$ as well as $(x, 1)$ with $(y, 1)$ and is the identity on all other literals. In total, the size of the symmetry group is 4. If we only allow the addition of ordering constraints, then any proper valid reduction of this CSP would actually be more symmetric. In particular, the symmetry ϕ cannot be removed through the addition of an ordering constraint, e.g., $y \leq x$. We show the microstructure of a valid reduction in Figure 3. The symmetry ψ that swaps $(x, 2)$ with $(y, 2)$ still exists in this valid reduction. Symmetry ϕ does not exist any more, but a symmetry that swaps $(y, 3)$ with $(y, 1)$. The size of the symmetry group of the valid reduction is 12. The symmetry group of this valid reduction is larger than the symmetry group of the original CSP.*

It has been noted before [Cohen *et al.*, 2006] that adding symmetry breaking constraints to eliminate *constraint* symmetries may result in a CSP that has symmetries that Cohen *et al.* call *solution symmetries* [Cohen *et al.*, 2006]. Example 9 shows that adding ordering constraints may augment the number of *constraint* symmetries as well, simply because in this example solution symmetries and constraint symmetries are the same. In Example 9, we can only eliminate the symmetries by removing nodes. We have depicted a valid reduction with a removed node in Figure 4. While this valid reduction does not contain an original symmetry or a restriction of an original symmetry, its symmetry group is as large as the symmetry group of the original CSP. Example 9 also shows that more than one valid reduction of a CSP may exist. As an example, in Figures 3 and 4 we depict the microstructures of two valid reductions of the CSP in Example 9.

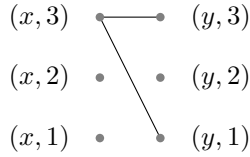


Figure 3: Adding $y \leq x$ to the CSP in Example 9 produces the above microstructure of a valid reduction with a larger symmetry group than the original CSP.

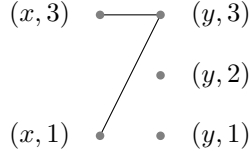


Figure 4: The microstructure of a tighter valid reduction of Example 9 having added $x \leq y$ and $x \neq 2$ to the CSP in Example 9.

In Definition 6, we do not require a valid reduction to be *strictly* tighter. Any CSP, symmetric or not, is a valid reduction of itself. Hence, we obtain the following generalisation of Puget’s original existence theorem for valid reductions for free.

Theorem 10 ([Puget, 1993]). *Every CSP has a valid reduction.*

When we reason about the existence of valid reductions, we must therefore be interested in showing the existence of a valid reduction that is strictly tighter than the original, symmetric CSP. We introduce this concept in the following definition.

Definition 11 (Proper Valid Reduction). *Let \tilde{P} be a valid reduction of a CSP P . If the MSC of \tilde{P} contains strictly fewer, not necessarily maximum, stable sets than the MSC of P , then we call \tilde{P} a proper valid reduction.*

We have seen in Example 1 that it is not always possible to find a proper valid reduction by adding an ordering constraint. We show a proper valid reduction of the CSP in Example 1 in Figure 1b obtained by adding a unary constraint. The empty CSP is an example of a CSP that is not symmetric according to Definition 5 and that does not allow a proper valid reduction.

The following theorem answers the question which symmetric CSPs admit a proper valid reduction.

Theorem 12. *A symmetric CSP has a proper valid reduction if and only if it has a partial solution that is not part of a solution or it has a symmetry ϕ that maps a solution S to a solution $\phi(S)$ such that $S \neq \phi(S)$.*

Proof. Let our symmetric CSP P have n variables and MSC G . A partial solution of P corresponds to a stable set in G .

(\Rightarrow) For the sake of contraposition, let us assume that in G each stable set is contained in a maximum stable set of size n and that all symmetries map any solution into itself. We cannot delete a node since that would destroy the only

candidate for the orbit representative in the orbit. We cannot add a hyperedge between any set of nodes that is a partial set: any such set of nodes is part of a solution by assumption and excluding this set of nodes would destroy the only candidate for the orbit representative. Hence, no proper valid reduction exists.

(\Leftarrow) We will show the existence of a proper valid reduction with MSC \tilde{G} . We remind the reader that according to Definition 5, P has at least one variable with domain size strictly larger than 1. Let P have n variables. The structure of this proof is a case differentiation. We will show that P admits a proper valid reduction with MSC \tilde{G} in each of the following cases:

1. P is unsatisfiable;
2. P is satisfiable and
 - (a) has one variable;
 - (b) has more than one variable and
 - i. a partial solution that is not contained in a solution;
 - ii. each partial solution is contained in a solution.

Case 1: P is not satisfiable. Hence, the MSC G does not admit a stable set of size n and therefore no orbit representatives of a stable set of size n need exist in the MSC of a valid reduction. By Definition 5, P has at least two literals that correspond to the same variable. Removing any such literal produces a proper valid reduction.

Case 2a: P is satisfiable and $n = 1$. Since any symmetric CSP has at least two literals by definition, the MSC G of P is then the complete graph on at least two nodes with the full permutation group acting on it. We can safely remove one node, no matter which one, to obtain an MSC of a proper valid reduction.

Case 2(b)i: P is satisfiable, $n > 1$ and a stable set exists that is not part of any maximum stable set. We can add a hyperedge in G consisting of the nodes in the stable set to obtain \tilde{G} .

Case 2(b)ii: We can exclude any stable set S that is not an orbit representative, by adding the n -ary hyperedge S to G to obtain \tilde{G} . \square

Theorem 12 singles out the symmetries that map a solution to a different solution as the important ones for valid reductions. However, before starting to prove the satisfiability of a CSP, we may only know that symmetries exist, e.g., through automatic detection in preprocessing. We cannot assume knowing whether symmetric solutions exist. Informally speaking, Theorem 12 points out symmetries that map every solution into itself as the badly-behaved symmetries.

The following is an example of a symmetric CSP with more than 1 solution that does not allow a proper valid reduction.

Example 13. *Consider a CSP with variables x, y and z each with domain $\{1, 2, 3\}$. The solutions are $\{(x, 1), (y, 1), (z, 1)\}$, $\{(x, 1), (y, 1), (z, 2)\}$, $\{(x, 1), (y, 2), (z, 3)\}$, and $\{(x, 2), (y, 1), (z, 2)\}$. The only edges in the microstructure are those that extend to these solutions. The microstructure of this CSP is depicted*

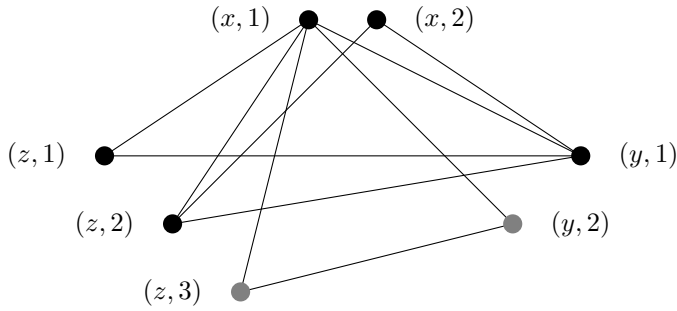


Figure 5: The microstructure of the CSP in Example 13 which does not admit a proper valid reduction. The only non-trivial symmetry of this CSP swaps the grey-coloured nodes $(y, 2)$ and $(z, 3)$ while being the identity on all other nodes. We can neither remove a node nor remove an edge.

in Figure 5. Apart from the identity, only one other automorphism exists. This automorphism swaps $(y, 2)$ with $(z, 3)$ and is the identity on all other literals. Note, that the automorphism maps all solutions into themselves. Every literal is part of a solution whose orbit consists only of itself and so are all pairs of nodes. Hence, we cannot remove a node or an edge. This means that no proper valid reduction of this CSP exists.

With another example, we show that we may not always need n -ary no-goods to obtain a proper valid reduction.

Example 14. Consider the microstructure of the CSP which is depicted in Figure 6. The edge $\{(x, 2), (y, 1)\}$ is dashed to facilitate identification. We have grouped the nodes in such a way that the symmetries of the microstructure are best visible. The graph is symmetric about an axis going through the nodes $(y, 1)$ and $(z, 1)$. Other symmetries of the CSP ensure that the 3-cliques to the left of the vertical axis going through nodes $(x, 1)$ and $(x, 2)$ are symmetric to each other and so are the 3-cliques to the right of this axis. The three orbits of the solutions under the symmetry group are as follows. The first orbit consists of $\{(x, 1), (y, 1), (z, a)\}$ and $\{(x, 2), (y, 2), (z, b)\}$, where $a \in \{4, 5\}$ and $b \in \{2, 3\}$. In the figure, this corresponds to the 3-cliques to the left of the vertical axis. The second orbit consists of $\{(x, c), (y, 1), (z, 1)\}$, where $c \in \{1, 2\}$. This corresponds to the two 3-cliques in the centre. The third and final orbit consists of the remaining 3-cliques to the right of the vertical axis. If we want to achieve a proper valid reduction of this CSP, without relying on a 3-ary no-good, we could remove the edge between nodes $(x, 2)$ and $(y, 1)$ which produces a proper valid reduction as well. In Figure 6, we have depicted this edge as dotted. This edge is contained in three members of the first orbit of solutions, but not in all. The edge is contained in one member of the second orbit of solutions, but not all. Removing this edge would destroy these four members, but the removal still leaves more than one orbit representative for the two affected orbits. Hence, the removal produces a proper valid reduction.

Stronger still, a unary constraint suffices in this example

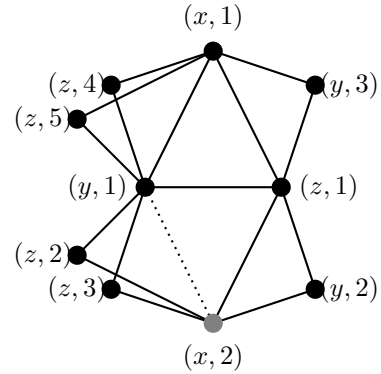


Figure 6: Microstructure of the CSP of Example 14. Removing the dotted edge or the grey node produces a proper valid reduction.

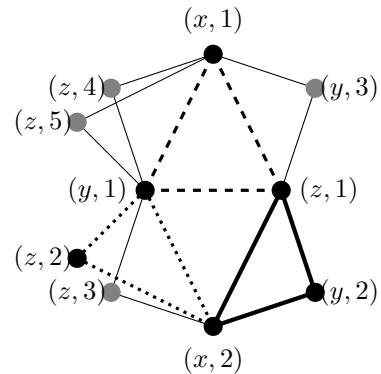


Figure 7: Microstructure of the CSP of Example 14. Orbit representatives are thick for $\{(x, 2), (y, 2), (z, 1)\}$, dotted for $\{(x, 2), (y, 1), (z, 2)\}$, and dashed for $\{(x, 1), (y, 1), (z, 1)\}$. We cannot remove a node or edge from non-orbit representative $\{(x, 2), (y, 1), (z, 1)\}$.

to obtain a proper valid reduction. By removing node $(x, 2)$, depicted in grey, we produce a proper valid reduction, since all stable sets above the axis through nodes $(y, 1)$, $(z, 1)$ are preserved.

In fact, we can show that Example 14 is an example for an extension of Theorem 12. If we can produce a proper valid reduction because a symmetry maps a solution to a different solution, we can always produce a proper valid reduction also by using constraints whose arity is at most binary.

Theorem 15. Let P be a symmetric CSP.

If P is unsatisfiable, then P has a proper valid reduction that can be obtained by adding unary constraints only.

If P is satisfiable and has a symmetry ϕ that maps a solution S to a solution $\phi(S)$ such that $S \neq \phi(S)$, then it has a proper valid reduction that can be obtained by adding binary constraints only.

Proof. Let the symmetric CSP P have n variables.

If P is unsatisfiable, we can prove the existence of a proper valid reduction with unary constraints relying on the argu-

ments used in Case 1 of the proof of Theorem 12. Let P be satisfiable in the following.

If $n = 1$ we can rely on the arguments used in Case 2a of the proof of Theorem 12. Let $n > 1$ in the following.

By assumption, at least one maximum stable set S exists that is mapped to a different stable set by some symmetry ϕ in $\text{aut}(G)$. Note that ϕ^{-1} is a symmetry since ϕ is a symmetry. Therefore $\phi^{-1}(T)$ is in the orbit of any maximum stable set T . By abuse of notation we write $v^{-1} = \phi^{-1}(v)$. Note that $|\phi(S) \setminus S| \neq \emptyset$ since $S \neq \phi(S)$.

Let v be any member of $\phi(S) \setminus S$ and let x be the variable which corresponds to this literal. Let w^{-1} be the unique member of S which also assigns a value to x . This literal w^{-1} exists, since S is an assignment of values to all variables. Note that $w^{-1} \neq v$ but that it may be possible that $v = w$. Let $T \supseteq \{v, w\}$ be any maximum stable set. For the sake of contradiction, we shall assume that each member of the orbit of T contains v . We know that T^{-1} is in the orbit of T since ϕ^{-1} is a symmetry. Since $w \in T$ it follows that $w^{-1} \in T^{-1}$. By our assumption that node v is contained in all members of the orbit of T , it follows that v and w^{-1} are both contained in T^{-1} . However, v and w^{-1} are literals corresponding to two different value assignments to x . By the definition of MSC, we have an edge between nodes v and w^{-1} , and this shows that T^{-1} cannot be a stable set. Having reached this contradiction, we can safely assume that any stable set $T \supseteq \{v, w\}$ has some member in its orbit which does not contain $\{v, w\}$. Therefore, we can safely forbid $\{v, w\}$ while preserving some orbit representative for the orbit of T . \square

Puget originally proved his existence theorem for valid reductions with a binary constraint. The proofs of Theorems 12 and 15 show that if a proper valid reduction exists because there are symmetric solutions, then we can obtain it by adding binary constraints only which is a strengthening of Puget's original result. However, we cannot use a technique as Puget used in his proof due to our Example 1.

3.3 Single-representative Valid Reductions and Beyond

Let us consider a particular class of valid reductions in the following. We call a valid reduction a *single-representative valid reduction*, if for each orbit of solutions in the original CSP only one solution exists in the single-representative valid reduction. Note that, if a symmetric CSP only has one solution or all solutions are only self-symmetric, a single-representative valid reduction is the original CSP. Example 13 shows a CSP that is a single-representative valid reduction of itself. Figures 3 and 4 show valid reductions that are proper valid reductions and single-representative valid reductions.

The proof of Theorem 12 leads to the following result.

Corollary 16. *Any symmetric CSP admits a single-representative valid reduction.*

Note that, if we have two distinct, symmetric solutions, a single-representative valid reduction is a proper valid reduction as well. In the proof of Theorem 12, we could have

proved Case 2(b)ii constructively with a reference to the generalised lexleader constraint [Walsh, 2006]. Using our proof of Case 2(b)ii however, we obtain a stronger result.

Corollary 17. *Any symmetric CSP admits a single-representative valid reduction for any choice of orbit representatives.*

An analogue of the above Corollary 17 for Theorem 15 would be that any symmetric CSP admits a single-representative valid reduction obtained with unary or binary constraints for any choice of orbit representatives. This is not true as the CSP in Example 14 shows. Consider the microstructure as in Figure 7. Let us choose $\{(x, 2), (y, 1), (z, 2)\}$ as the orbit representative of the first orbit of solutions to the left of the axis going through $(x, 1)$ and $(x, 2)$. The edges of this 3-clique are depicted as dotted lines. Let the 3-clique $\{(x, 1), (y, 1), (z, 1)\}$ with dashed lines as edges be the orbit representative of the second, inner orbit and let the 3-clique $\{(x, 2), (y, 2), (z, 1)\}$ with thick lines as edges be the orbit representative of the third orbit of solutions to the right of the axis going through $(x, 1)$ and $(x, 2)$. We cannot remove an edge or a node from the non-orbit-representative 3-clique in the second orbit $\{(x, 2), (y, 1), (z, 1)\}$. Hence, with this choice of orbit representatives and restricting ourselves to unary and binary constraints for producing a single-representative valid reduction, no single-representative valid reduction exists.

Generalised lexleader constraints [Walsh, 2006] have the same arity as the number of variables in general. For some CSPs however, we can reduce the arity of lexleader constraints to 2 [Puget, 2005]. As Example 14 shows, we cannot rely on binary constraints only to produce a single-representative valid reduction for an arbitrary choice of orbit representatives. It remains an open question whether for a CSP with n variables and constraints with arity at most $k < n$, we can obtain a single-representative valid reduction by adding k -ary constraints only.

Since the definition of a single-representative valid reduction focuses on excluding symmetric solutions, and does not specifically exclude symmetric non-solutions, we may still be able to exclude further sets of literals from a single-representative valid reduction. Informally speaking, a single-representative valid reduction may not be the most proper valid reduction. In future work, we will consider symmetric non-solutions and theoretical conditions for the exclusion of symmetric non-solutions via constraints.

In practice, we cannot distinguish between orbits of solutions and orbits of non-solutions. Lexicographic ordering constraints [Crawford *et al.*, 1996] exclude orbit members of both solutions and non-solutions. We can produce a single-representative valid reduction with generalised lexleader constraints [Walsh, 2006] constructively. We sacrifice free choice of orbit representatives in this case. Generalised lexleader constraints, however, do not guarantee a single-representative valid reduction that does not exhibit any symmetry of the original CSP. We have seen so in Examples 1 and 9: certain types of symmetries (not symmetric solutions) exist that we cannot eliminate using lexicographic ordering constraints or generalised lexleader constraints. The reason is that adding

lexicographic ordering constraints does not change the MSC in these cases. Only unary constraints can eliminate the symmetries in these examples. However, we are not aware of a constructive method that provides unary constraints to produce a single-representative valid reduction. Future work will address the question whether there are less pathological examples where unary constraints are needed to eliminate symmetries. The existence of such an example would mean that we would have to understand the interplay between unary and lexicographic ordering constraints. In general, as shown in Example 13, we know that we cannot always produce a single-representative valid reduction with all symmetries eliminated. In future work, we will consider conditions under which we can guarantee a single-representative valid reduction with all symmetries eliminated.

4 Conclusion

We have presented Puget's fundamental work [Puget, 1993] on symmetry breaking in light of the recent definition of symmetry [Cohen *et al.*, 2006]. We have re-proved his central results and strengthened them. Our results suggest that symmetry breaking based on the idea of valid reductions may, in theory, only supply additional information to a problem if symmetric solutions exist. This is somewhat disappointing, since the existence of symmetric solutions can be decided only by solving the CSP.

Valid reductions are among the requirements for a constraint to be considered as a symmetry breaking constraint. Hence, they offer a possibility to provide a theoretical upper bound for the power of symmetry breaking constraints. While our proofs suggest that valid reductions allow for constraints that are far from our intuition about symmetry breaking constraints, valid reductions provide theoretical limits for symmetry breaking constraints and offer a different perspective of symmetry breaking constraints. An interesting open question is whether we could theoretically have constraints that lead to single-representative valid reductions that have the same arity as the maximum constraint arity in a CSP. In further work, we will analyse and compare the very diverse symmetry breaking constraints using valid reductions as the framework.

Acknowledgements We would like to thank Timo Berthold, Haiko Müller, Marc E. Pfetsch and Kristina Vušković for helpful discussions on the topic.

References

[Cambazard *et al.*, 2009] Hadrien Cambazard, Barry O'Sullivan, and Barbara M. Smith. A Constraint-Based Approach to Enigma 1225. *Journal of Computers and Mathematics with Applications*, 2009.

[Cohen *et al.*, 2006] David Cohen, Peter Jeavons, Christopher Jefferson, Karen Petrie, and Barbara Smith. Symmetry definitions for constraint satisfaction problems. *Constraints 11*, 2006.

[Crawford *et al.*, 1996] James Crawford, Matthew Ginsberg, Eugene Luks, and Amitabha Roy. Symmetry-breaking

predicates for search problems. In *Principles of Knowledge Representation and Reasoning (KR '96)*. Morgan Kaufmann, 1996.

- [Flener *et al.*, 2006] Pierre Flener, Justin Pearson, Meinolf Sellmann, and Pascal Van Hentenryck. Static and dynamic structural symmetry breaking. In *Principles and Practice of Constraint Programming – CP 2006*, volume 4204/2006 of *Lecture Notes in Computer Science 4204*, 2006, pages 695–699, 2006.
- [Freuder, 1991] E. C. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings AAAI'91*, pages 227–233, 1991.
- [Jégou, 1993] Philippe Jégou. Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In *Proceedings AAAI*, pages 731–736, 1993.
- [Kaibel and Pfetsch, 2008] Volker Kaibel and Marc E. Pfetsch. Packing and partitioning orbitopes. *Mathematical Programming*, 114, Number 1 / July, 2008:1–36, 2008.
- [Puget, 1993] Jean-Francois Puget. On the satisfiability of symmetrical constrained satisfaction problems. In *Methodologies for Intelligent Systems*, pages 350–361, London, UK, 1993. Springer-Verlag.
- [Puget, 2005] Jean-Francois Puget. Breaking symmetries in all-different problems. In *International Joint Conferences on Artificial Intelligence*, pages 272–277, 2005.
- [Rossi *et al.*, 2006] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *Handbook of Constraint Programming*. Elsevier, 2006.
- [Shlyakhter, 2007] Ilya Shlyakhter. Generating effective symmetry-breaking predicates for search problems. *Discrete Applied Mathematics*, 155(12):1539 – 1548, 2007.
- [Walsh, 2006] Toby Walsh. General symmetry breaking constraints. In *Principles and Practice of Constraint Programming – CP 2006*, volume 4204/2006 of *Lecture Notes in Computer Science 4204*, 2006, pages 650–664, 2006.