Ann Math Artif Intell (2009) 57:75–102
DOI 10.1007/s10472-010-9174-1

Minimal ordering constraints for some families of variable symmetries

Andrew Grayland · Chris Jefferson · Ian Miguel · Colva M. Roney-Dougal

Published online: 3 March 2010

Abstract Variable symmetries in a constraint satisfaction problem can be broken by adding lexicographic ordering constraints. Existing general methods of generating such sets of ordering constraints can require a huge number of constraints. This adds an unacceptable overhead to the solving process. Methods exist by which this large set of ordering constraints can be reduced to a much smaller set automatically, but their application is also prohibitively costly. In contrast, this paper takes a bottom-up approach. It examines some commonly-occurring families of groups and derives a minimal set of ordering constraints sufficient to break the symmetry each group describes. These minimal sets are then used as building blocks to generate minimal sets of ordering constraints for groups constructed via direct and imprimitive wreath products. Experimental results confirm the value of minimal sets of ordering constraints, which can now be generated much more cheaply than with previous methods.

Keywords Constraint programming · Constraint modelling · Symmetry

Mathematics Subject Classifications (2000) Primary 68T20; Secondary 20B25 · 20B40

A. Grayland (✉) · C. Jefferson · I. Miguel
School of Computer Science, University of St Andrews, Fife, UK
e-mail: andyg@cs.st-andrews.ac.uk

C. Jefferson
e-mail: caj@cs.st-andrews.ac.uk

I. Miguel
e-mail: ianm@cs.st-andrews.ac.uk

C. M. Roney-Dougal
School of Maths and Statistics, University of St Andrews, Fife, UK
e-mail: colva@mcs.st-andrews.ac.uk

1 Introduction

Constraint programming supports the solution of a combinatorial problem in two stages. First, the problem is characterised or *modelled* as a *constraint satisfaction problem* (CSP): a finite set of decision variables, each with a finite set of potential values, and a set of constraints on the allowed assignments of values to the variables. Second, a constraint solver is used to search for *solutions*: assignments to the decision variables that satisfy all the constraints. Constraint models often contain *symmetries* that partition the set of assignments into equivalence classes. Symmetries can be exploited by restricting the search to one member (or a reduced number of members) of each equivalence class (*symmetry breaking*), dramatically reducing search.

One symmetry-breaking method is to add constraints to the model. Crawford et al. [2] describe one such approach called *lex-leader*: one member of each equivalence class is designated as lexicographically least, and a set of lexicographic ordering constraints are added to preclude all other members of that class. The disadvantage of this method is that, for a CSP with n variables, it produces up to $n!$ lexicographic ordering constraints. The overhead of adding this number of constraints to the CSP usually outweighs the benefit of breaking the symmetry.

In many cases, this large set of constraints can be reduced to a much smaller set that still breaks all the symmetry [4, 6]. However, these general reduction methods are themselves prohibitively costly. Puget [9] has identified a special case, where each variable must be assigned a distinct value, in which the set of ordering constraints collapses to just $n-1$ binary inequalities. This paper follows somewhat in this vein. It considers the mathematical *group* that describes the set of all variable symmetries, the associated set of lexicographic ordering constraints necessary to break those symmetries, and how that set can be reduced to a minimal fixpoint.

Further to this, the paper discusses ways of combining commonly occurring groups: we concentrate on the combinations that occur most frequently in CSPs. These combinations are much larger than the groups they are constructed from; they therefore pose a similar problem in relation to the number of constraints required to break the symmetries in a CSP with that symmetry group. We examine methods of breaking these symmetries by utilising the minimal sets of constraints defined for the constituent groups.

2 Background

A finite-domain constraint satisfaction problem comprises: a finite set of variables $\mathcal{X}$; for each variable $x \in \mathcal{X}$, a finite set of values (its *domain*); and a finite set $\mathcal{C}$ of constraints on the variables. Each constraint $c \in \mathcal{C}$ is defined over a sequence, $\mathcal{X}'$, of variables drawn from $\mathcal{X}$. A subset of the Cartesian product of the domains of the members of $\mathcal{X}'$ gives the set of allowed combinations of values. A *complete assignment* maps every variable in a given CSP to a member of its domain.

A *variable symmetry* of a CSP is a bijection $f : \mathcal{X} \rightarrow \mathcal{X}$ of the set of variables such that $\{\langle x_i, a_i\rangle \ : \ 1 \leq i \leq n\}$ is a solution if and only if $\{\langle f(x_i), a_i\rangle \ : \ 1 \leq i \leq n\}$ is a solution.

Any group can be represented by a set G of bijections from a set $\mathcal{X}$ to itself (or *permutations* of the set $\mathcal{X}$), such that G is closed under composition of functions and inversion. The groups we are interested in are sets of variable symmetries of the CSP. The symmetric group, S_n, is the group whose elements are the set of all possible bijections from $\{1, \ldots, n\}$ to itself.

Having identified a symmetry within a model, we identify a set of symmetry-breaking constraints sufficient to break it using the lex-leader method. We first define an ordering on the decision variables,[1] then add constraints that order the assignments to these variables. To illustrate, we consider the symmetric group S_3 acting on three variables, x_1, x_2, and x_3. Here the permutation $[x_1, x_3, x_2]$ means x_1 maps to itself, and x_2 and x_3 map to each other:

$$[x_1, x_2, x_3], [x_2, x_1, x_3], [x_1, x_3, x_2], [x_3, x_2, x_1], [x_2, x_3, x_1], [x_3, x_1, x_2]$$

We choose x_1 to be the most significant variable in the ordering, x_2 the next most significant, and x_3 the least significant. The next step is to add symmetry-breaking constraints to allow only one member of each equivalence class of assignments induced by the symmetry:

$$x_1x_2x_3 \leq_{\text{lex}} x_2x_1x_3,\ \ x_1x_2x_3 \leq_{\text{lex}} x_3x_2x_1,\ \ x_1x_2x_3 \leq_{\text{lex}} x_3x_1x_2$$

$$x_1x_2x_3 \leq_{\text{lex}} x_1x_3x_2,\ \ x_1x_2x_3 \leq_{\text{lex}} x_2x_3x_1$$

Notice that there is one constraint per nontrivial permutation in S_3. In general, there are $n!$ permutations for the group S_n and so $(n! - 1)$ n-ary constraints are produced by the lex-leader method to break all symmetries.

Every permutation can be written as a composition of disjoint cycles. Consider the following permutation f in list notation:

$$f = [x_3, x_1, x_6, x_4, x_7, x_2, x_8, x_5].$$

We see that $f(x_1) = x_3$, $f(x_3) = x_6$, $f(x_6) = x_2$ and $f(x_2) = x_1$; similarly, $f(x_5) = x_7$, $f(x_7) = x_8$ and $f(x_8) = x_5$, while $f(x_4) = x_4$. In cycle notation

$$f = (x_1x_3x_6x_2)(x_4)(x_5x_7x_8).$$

We generally omit mappings of elements to themselves; this is the notation we will use to represent permutations in the rest of this paper.

$$f = (x_1x_3x_6x_2)(x_5x_7x_8)$$

3 Reducing the ordering constraints

Rules 1 and 2 were introduced in [4] to reduce the number and arity of lex constraints whilst maintaining logical equivalence. Rule 3, which supercedes and is stronger than

[1]Note that recent research suggests that the ordering chosen can affect the search tree quite considerably [10].

Rules 1 and 2, is defined in [6]. Let α, β, γ, and δ be strings of variables, and x and y be individual variables.

1. If $\alpha = \gamma$ entails $x = y$ then a constraint c of the form $\alpha x \beta \leq_{\text{lex}} \gamma y \delta$ may be replaced with $\alpha\beta \leq_{\text{lex}} \gamma\delta$.
2. If $C = C' \cup \{\alpha x \leq_{\text{lex}} \gamma y\}$ is a set of constraints, and $C' \cup \{\alpha = \gamma\}$ entails $x \leq y$, then C may be replaced with $C' \cup \{\alpha \leq_{\text{lex}} \gamma\}$.
3. If C is a set of constraints of the form $C' \cup \{\alpha x \beta \leq_{\text{lex}} \gamma y \delta\}$, and $C' \cup \{\alpha = \gamma\}$ entails $x = y$ (or $C' \cup \{\alpha = \gamma\}$ entails $x \leq y$ where $|\beta| = 0$), then C may be replaced with $C' \cup \{\alpha\beta \leq_{\text{lex}} \gamma\delta\}$.

For example, consider the constraint $x_1x_2 \leq_{\text{lex}} x_2x_1$. From the definition of lexicographic ordering, to ensure that the constraint is satisfied we need only compare a pair of variables if each pair of more significant variables are equal. Here, if $x_1 = x_2$ then trivially the second pair *must* be equal. Therefore, by Rule 1 we need only consider the first pair of variables, reducing this constraint to $x_1 \leq x_2$ without modifying the set of solutions. In the S_3 example given in the previous section, Rule 1 reduces the set of lexicographic ordering constraints to: $x_1 \leq x_2,\ x_2 \leq x_3,\ x_1 \leq x_3,\ x_1x_2 \leq_{\text{lex}} x_2x_3,\ x_1x_2 \leq_{\text{lex}} x_3x_1$. Application of Rule 2 simplifies the constraints further to: $x_1 \leq x_2,\ x_2 \leq x_3$.

Rule 3 supercedes and is stronger than Rules 1 and 2. Rule 3 extends Rules 1 and 2 by allowing both the consideration of all pairs of variables in any one lex constraint, provided by Rule 1, and the implications derived from considering the entire set of lex constraints, provided by Rule 2. Unfortunately the support required for removal of the least significant pair remains essentially different from that required for the removal of any other pair. We can remove any least significant pair in a lex constraint by showing that the inequality that it states is entailed whenever it is considered. However, to remove any other pair we must show that its variables are equal at the time it is considered. For this reason we find it useful to remove the action of Öhrman's Rule 3 that covers the actions of Rule 2 and to restate it as Rule 3′.

3′ If C is a set of constraints of the form $C' \cup \{\alpha x \beta \leq_{\text{lex}} \gamma y \delta\}$, and $C' \cup \{\alpha = \gamma\}$ entails $x = y$, then replace C with $C' \cup \{\alpha\beta \leq_{\text{lex}} \gamma\delta\}$.

4 Confluence and minimality

We now show that the application of Rule 1 is *confluent*, i.e., there exists a unique fixpoint in its application. We also show that in general the same is not true of Rule 2 nor Rule 3′.

Lemma 1 *Application of Rule 1 is confluent.*

Proof Consider a pair of variables at an index i that can be removed by Rule 1. The justification for this removal is a subset of the pairs of variables at the indices less than i. Assume that Rule 1 is not confluent. Hence, for some pair at index i, there exists an index h, where $h < i$, at which part of the justification for the removal of the pair at i has been removed by a previous application of Rule 1. However, by definition the equality of the pair at index h is entailed by a subset of the pairs at indices less than h. Hence, the 'hole' in the justification for i can be repaired by the justification for h. Of course, the pair at h may also have a hole in its justification, but this hole can be repaired in the same way. Furthermore, there must exist a leftmost pair of variables that can be removed by Rule 1 whose justification cannot have a hole. This is a contradiction. □

Lemma 2 *Application of Rule* 2 *is not confluent.*

Proof Consider the following set of lexicographic ordering constraints:

$$x_1x_2x_3x_5 \leq_{\text{lex}} x_2x_3x_4x_6$$

$$x_1x_2 \leq_{\text{lex}} x_3x_4$$

$$x_1x_2x_3x_5 \leq_{\text{lex}} x_4x_1x_2x_6$$

Rule 2 can now be applied in one of two ways.

1. Consider $x_1x_2x_3x_5 \leq_{\text{lex}} x_2x_3x_4x_6$, with $x_5 \leq x_6$ for removal. We can apply Rule 2 to a fixpoint, leaving:

$$x_1 \leq_{\text{lex}} x_2$$

$$x_1x_2 \leq_{\text{lex}} x_3x_4$$

$$x_1x_2x_3x_5 \leq_{\text{lex}} x_4x_1x_2x_6$$

2. Consider $x_1x_2x_3x_5 \leq_{\text{lex}} x_4x_1x_2x_6$, with $x_5 \leq x_6$ for removal. We can apply Rule 2 to a fixpoint leaving:

$$x_1x_2x_3x_5 \leq_{\text{lex}} x_2x_3x_4x_6$$

$$x_1x_2 \leq_{\text{lex}} x_3x_4$$

$$x_1x_2x_3 \leq_{\text{lex}} x_4x_1x_2$$

□

Lemma 3 *Application of Rule* 3′ *is not confluent.*

Proof Consider the following set of lexicographic ordering constraints:

$$x_1x_{10}x_3 \leq_{\text{lex}} x_2x_{11}x_4$$

$$x_1x_{10}x_5 \leq_{\text{lex}} x_2x_{11}x_6$$

$$x_1x_{11}x_7 \leq_{\text{lex}} x_2x_{10}x_8$$

Rule 3′ can now be applied in one of two ways.

1. Consider $x_1x_{10}x_3 \leq_{\text{lex}} x_2x_{11}x_4$ with $x_{10} \leq x_{11}$ for removal. We can apply Rule 3′ to a fixpoint leaving:

$$x_1x_3 \leq_{\text{lex}} x_2x_4$$

$$x_1x_{10}x_5 \leq_{\text{lex}} x_2x_{11}x_6$$

$$x_1x_{11}x_7 \leq_{\text{lex}} x_2x_{10}x_8$$

2. Consider $x_1x_{10}x_5 \leq_{\text{lex}} x_2x_{11}x_6$ with $x_{10} \leq x_{11}$ for removal. We can apply Rule 3′ to a fixpoint leaving:

$$x_1x_{10}x_3 \leq_{\text{lex}} x_2x_{11}x_4$$

$$x_1x_5 \leq_{\text{lex}} x_2x_6$$

$$x_1x_{11}x_7 \leq_{\text{lex}} x_2x_{10}x_8$$

□

Definition 1 A set of lexicographic ordering constraints is said to be Rule 2, 3′ minimal if the set is unchanged under application of Rules 2 and 3′.

We believe that a Rule 2 and 3′ minimal set of lexicographic ordering constraints will outperform an equivalent complete set of lex constraints. Achieving minimality from a factorial number of constraints by mechanical application of Rules 2 and 3′ is expensive. In practice, it has proved infeasible as much of the time saved by breaking the symmetries is re-introduced in this pre-processing stage [6]. We can derive a lower bound on the number of binary inequality constraints produced by this process.

A group of variable symmetries is *transitive* if any variable can be mapped to any other.

Theorem 1 *Let $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ be the set of decision variables of a CSP P. Assume that P has a transitive group of variable symmetries and that each domain has size at least 2. Then, not considering any other constraints in P, the minimum number of binary $\leq$ constraints required to remove all but one member of each equivalence class of assignments is $n-1$.*

Proof Any constraint graph with $n-2$ binary constraints is disconnected. Let x_1 be the most significant variable. Let x_i be a decision variable that is not connected to x_1, and such that x_i is most significant in its component of the constraint graph. Let a be the minimal element of the domain of x_i (and hence of all variables, by transitivity). Let A be any full assignment which assigns $x_i = a$ and all other variables to values other than a. Since there is a symmetry mapping x_i to x_1 there is a symmetry mapping A to a full assignment with $x_1 = a$. Thus there will remain more than one solution from an equivalence class after addition of the $n-2$ binary constraints, and therefore $n-2$ binary constraints will not suffice. Puget [9] has described a set of binary $\leq$ constraints of size $n-1$ that break all symmetries in some cases. The minimum number is therefore $n-1$. □

5 Reduction rules and inequality graphs

In the previous section we described Rules 1, 2 and 3′ and the concept of minimality of a set of lexicographic ordering constraints. We begin by introducing the *inequality graph* as a device to visualise the application of Rules 1, 2, and 3′. The inequality graph was first introduced by Öhrman [6]. We describe it's function here and give new insights into its mechanics.

Definition 2 Given a CSP with set of variables $\mathcal{X}$ and set of constraints $\mathcal{C}$, the corresponding inequality graph G is a directed graph with one node per element of $\mathcal{X}$. The edges of G are as follows: if, for some $x_i, x_j \in \mathcal{X}$, the set $\mathcal{C}$ contains:

1. $x_i \leq x_j$ then G contains a directed edge from x_i to x_j.
2. $x_i = x_j$ then G contains a directed edge from x_i to x_j, and a directed edge from x_j to x_i.
3. $x_i \ldots \leq_{\text{lex}} x_j \ldots$ then G contains a directed edge from x_i to x_j.

Given a set of lexicographic constraints $\mathcal{C}$ on variables $\mathcal{X}$, we consider the process of applying Rule 3′ to remove the least significant pair of variables in some $c \in \mathcal{C}$. Following the rule, we begin by adding equality constraints between all more significant pairs of variables in c. To illustrate, Fig. 1 shows the inequality graph for a CSP where $\mathcal{X} = \{A, B, C, D\}$, and $\mathcal{C} = \{ABC \leq_{\text{lex}} BCD, ABC \leq_{\text{lex}} CDA\}$. Here the pair $C \leq D$ from the constraint $ABC \leq_{\text{lex}} BCD$ is under consideration for removal by Rule 2. The assumed equalities, $A = B$ and $B = C$, are represented by pairs of directed edges between the respective nodes and the inequality $A \leq C$ entailed by $ABC \leq_{\text{lex}} CDA$ is represented by a directed edge from A to C.

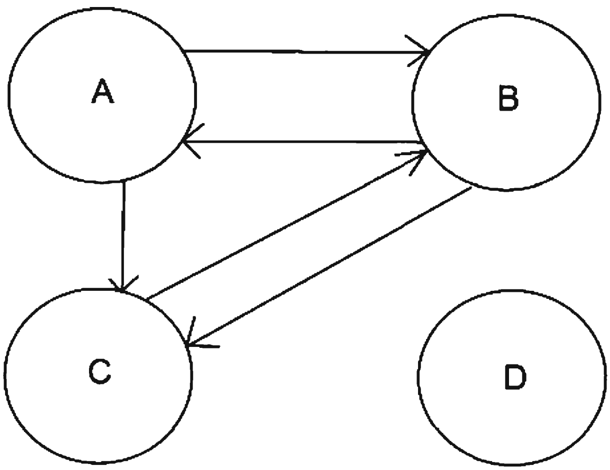


Fig. 1 An inequality graph for the application of Rule 2 to remove $C \leq D$ from $ABC \leq_{\text{lex}} BCD$ in the context of $ABC \leq_{\text{lex}} CDA$

The antecedent of Rule 2 requires the identification of entailed inequality constraints. The process of identifying such equalities can be characterised partially in terms of taking the transitive closure of the inequality graph.

Definition 3 Let $G = (\mathcal{X}, E)$ be a directed graph. The *transitive closure* of G is a graph denoted $G^t = (\mathcal{X}, E')$ such that for all $x_i, x_j \in \mathcal{X}$ there is an edge (x_i, x_j) in E' if and only if there is a path from x_i to x_j in G [7].

Returning to our example, since there is a path from C to A via B, taking the transitive closure of the inequality graph in Fig. 1 adds a directed edge from C to A. Since there is now both a directed edge from A to C, and from C to A, we can deduce $A = C$. This allows us to simplify the constraint $ABC \leq_{\text{lex}} CDA$ to $BC \leq_{\text{lex}} DA$. Consequently, it is now clear that $B \leq D$. Hence, the inequality graph of this new, simpler problem contains a directed edge from B to D. The transitive closure of this graph is shown in Fig. 2. Notice that there is a directed edge from C to D, hence the pair C and D can be removed from $ABC \leq_{\text{lex}} BCD$ via Rule 2.

Generally, as pointed out by Öhrman [6], Rule 2 can be applied as follows.

1. Let P be the initial CSP, combined with the equality constraints assumed by Rule 2.
2. Let G^t be the transitive closure of the inequality graph of P.
3. If G^t contains edges corresponding to equalities not represented explicitly in P, add these equalities to P and go to 2.
4. Otherwise, the antecedent of Rule 2 is satisfied if the corresponding edge is present in G^t.

The following lemma characterises the situations in which equality between two decision variables is entailed in the context of a CSP containing lexicographic ordering and equality constraints.

Lemma 4 *Given a CSP with set of variables $\mathcal{X}$ and set of constraints $\mathcal{C}$, containing equality and lexicographic ordering constraints, further equality constraints between*

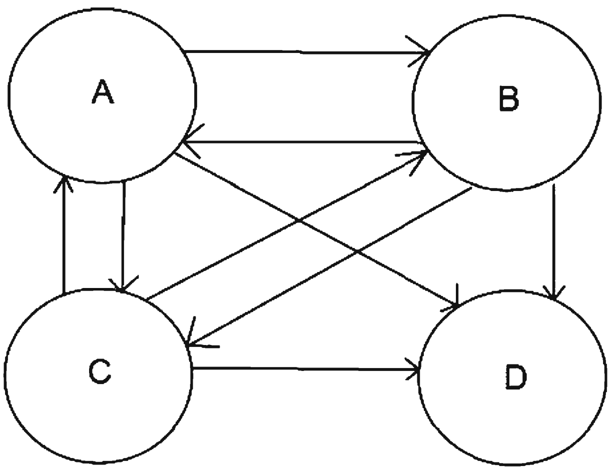


Fig. 2 Transitive closure of the inequality graph for a CSP with $\mathcal{X} = \{A, B, C, D\}$, $\mathcal{C} = \{A = B, A = C, B = C, BC \leq_{\text{lex}} DA\}$.

pairs of variables $x_a, x_b \in \mathcal{X}$ are entailed if and only if at least one of the following holds:

1. *(transitivity): there exists some $x_c \in \mathcal{X}$ (distinct from x_a and x_b) such that $x_a = x_c$ and $x_b = x_c$; or*
2. *$x_a \leq x_b$ and $x_b \leq x_a$; or*
3. *there exists some $x_c \in \mathcal{X}$ (distinct from x_a and x_b) such that $x_a \leq x_b \leq x_c$ and $x_c \leq x_a$ (and similarly, exchanging the roles of x_a, x_b).*

Proof It is clear that if one of the three possibilities occurs, then $x_a = x_b$, so we must prove the converse.

Let G be the inequality graph of a CSP whose decision variables include x_a and x_b. Assume that it is possible to apply that $x_a = x_b$. We wish to show that one of the three given cases occur.

To show that $x_a = x_b$ there must be a directed edge from x_a to x_b and a directed edge from x_b to x_a in G^t. Let $G' = G - \{x_a, x_b\}$ and let G_1 be G' plus the nodes x_a and x_b, with the same edges incident to x_a or x_b as in G. Then G_1 is a partial transitive closure of G.

There are three possibilities: firstly that $x_a \leq x_b$ is in G, secondly that $x_b \leq x_a$ is in G, and thirdly that there is no initial inequality between x_a and x_b. For each we must show that at least one of the cases listed in the theorem occurs.

Consider first the possibility that $x_a \leq x_b$, and so there is a directed edge from x_b to x_a in G_1. If in G_1 there are only directed edges from x_b to G'^t there is no path from x_a to x_b in G_1, contradicting our assumption that we can prove that $x_a = x_b$. Similarly, if in G_1 there are only directed edges from G'^t to x_a then the same holds. There therefore exists at least one edge directed from x_a to G'^t and one edge directed to x_b from G'^t. Let (x_a, x_i) and (x_j, x_b) be directed edges in G_1, with $x_i, x_j \in G'^t$. There exists some path from x_i to x_j in G'^t to show that $x_a = x_b$. We can then, without loss of generality, consider a directed edge from x_a to x_i and a directed edge from x_i to x_b, since we will ultimately be working in the transitive closure. This represents the equality $x_a \leq x_i \leq x_b$. Since $x_a \leq x_b$ was already present, this is an example of case 2.

The second case, where we start with $x_b \leq x_a$, follows by symmetry.

Consider now the third and final case where there is no directed edge in G between x_a and x_b or x_b and x_a. In G_1 there must be paths from x_a to x_b and vice versa, since in G^t there are edges from x_a to x_b and vice versa. Where the two paths are distinct the same reasoning as with the previous case gives $x_a \leq x_b$ for one path and $x_b \leq x_a$ for the other, therefore $x_a = x_b$. So suppose instead that the two paths follow the same nodes in opposite directions. Let x_c be on the path, then without loss of generality, we consider directed edges from x_a to x_c and x_c to x_a, and directed edges from x_c to x_b and x_b to x_c. This represents the equations $x_a = x_c$ and $x_b = x_c$, therefore $x_a = x_b$. □

Having shown the ways in which equality can be deduced within lex constraints we now make an observation about the application of Rule 3′.

Corollary 1 *Given a set of lexicographic ordering constraints, a pair of variables which are not the least significant pair in that particular constraint can be discarded using Rule* 3′ *if and only if they are shown to be equal by Lemma* 4 *or the pair was assumed to be equal under the actions of Rule* 3′.

Additionally we note a relationship between the deductions from assumptions over a set of variables and the possible deductions from assumptions over a proper subset of these variables.

Corollary 2 *Let $\chi = \{x_1, \ldots, x_n\}$ be a set of variables, and let L and C be a set of lexicographic and equality constraints on χ, respectively. Let E be the set of equalities deduced using Lemma* 4 *from L and C. Let $C' \subset C$, then the set of equalities deduced using Lemma* 4 *from L and C' is a subset of E.*

We also establish a useful pattern in the simplification of a set of lexicographic ordering constraints.

Lemma 5 *Given a set of lexicographic ordering constraints $\mathcal{C}$, and some $c \in \mathcal{C}$ of the form $x_1 \ldots x_i \leq_{\text{lex}} y_1 \ldots y_i$, the pair x_i, y_i can be discarded from c if and only if $x_i \leq y_i$ is entailed by the assumptions: $x_1 = y_1$, $x_2 = y_2$, …, $x_{i-1} = y_{i-1}$. This implication requires x_i, or a variable assumed to be equal to x_i, to appear on the left-hand side of some $c' \in \mathcal{C}$, where $c \neq c'$.*

Proof Consider an inequality graph, G, of any problem involving the decision variable x_i. Consider also the graph G', where $G' = G - \{x_i\}$. Adding x_i to G'^t with the same edges incident with x_i as in G gives a partial transitive closure graph G_1 of G.

Assume first of all that x_i has no edges to any other node in G_1. Then x_i is disconnected in G_1^t, therefore the pair under consideration cannot be removed.

Assume now that there exists directed edges only from x_i to G'^t in G_1, so that x_i is less than or equal to some other variables. The transitive closure of G_1 can only add edges from x_i to other nodes in G, since it is only possible to find a path away from x_i, and never to x_i. If there is no path from any node in G to x_i, there can be no edge from y_i to x_i, and the pair x_i, x_j cannot be removed.

The same idea can then be extended to larger sets of nodes. If $x_i = x_j$, there must exist an edge from G'^t to either x_i or x_j in order to show that x_i is less than another node in G. □

Lemma 6 *The most significant pair in any lex constraint can be removed using Rule $3'$ if and only if it is a relation constraining one variable against itself.*

Proof When applying Rule $3'$ to the most significant pair $x_1 \leq x_i$ of any lex constraint we make no assumptions of equality. In order to remove the most significant pair in a constraint by Rule $3'$ we must show that x_1 and x_i are equal.

If a pair contains x_1 twice over this is trivially true.

Where $j \neq i$ there must be some other inequalities which force equality between these two variables. There are no equalities entailed within any lex constraints if no assumptions are made, so no lex constraint can have x_1 on the right hand side of its most significant pair if it is not also on the left hand side of the pair. Thus we cannot prove that any variable is less than or equal to x_1. □

6 Minimal sets of lexicographic ordering constraints for some families of groups

We now consider some specific families of groups. In each case we produce a number of lex constraints that is linear in the number of variables, and show that our new set of constraints is logically equivalent to the full set of lex-leader constraints. Where possible, we also show that our new set of constraints is minimal.

6.1 Symmetric groups

Recall that the symmetric group, S_n, is the group whose elements are the set of all bijections from $\{1, \ldots, n\}$ to itself. Symmetric groups arise frequently as symmetries of CSPs, in particular whenever a set is modelled as a list we introduce the symmetric group on those variables.

Theorem 2 *Given a CSP with n decision variables* $\{x_1, x_2, \ldots, x_n\}$ *whose symmetry group is* S_n *on variables, a complete set of symmetry breaking constraints is:*

$$x_i \leq x_{i+1} \ \textit{for}\ 1 \leq i \leq n-1$$

Proof We first show that the complete set of lex-leader constraints entail our constraints, then show that the reduced set of constraints entails the lex-leader constraints. Since the lex-leader constraints are complete, the reduced set of constraints are complete.

Lex-leader constraints entail the reduced set of constraints, as the reduced set is a subset of the lex-leader constraints.

The reduced set of constraints entails the complete set of lex-leader constraints since the lex-leader constraints break every permutation of S_n, which is every possible permutation of n variables. The reduced set implies that $x_1 \ldots x_n$ is sorted, which breaks every possible permutation of n variables. □

Theorem 3 *The reduced set of symmetry breaking constraints for* S_n *is minimal.*

Proof Theorem 1 states that the minimum number of binary inequalities required to break symmetry for a transitive group is $n-1$, where n is the number of variables. There are $n-1$ inequalities in the reduced set of symmetry breaking constraints for S_n therefore they cannot be reduced further by Rules 2 or 3' and are minimal. □

6.2 Cyclic groups

If all elements of a group G can be written as powers of some fixed $g \in G$ then G is *cyclic*. In this subsection we produce a set of minimal lex constraints for breaking the cyclic group. We will assume throughout that the elements of the cyclic group are powers of the permutation $(x_1, x_2, \ldots, x_n)$.

Theorem 4 *Let P be a CSP with n decision variables,* $\{x_1, x_2, \ldots, x_n\}$*. If the symmetry group of P is a cyclic group of variable symmetries then a complete set* $\mathcal{M}$ *of symmetry breaking constraints is:*

$$\begin{aligned} x_1 &\leq x_2 \\ x_1x_2 &\leq_{\text{lex}} x_3x_4 \\ x_1x_2x_3 &\leq_{\text{lex}} x_4x_5x_6 \\ &\vdots \\ x_1x_2\ldots x_{n/2+1} &\leq_{\text{lex}} x_{n/2+2}\ldots x_nx_1x_2 \quad n \text{ even} \\ x_1x_2\ldots x_{(n-1)/2}x_{(n+1)/2} &\leq_{\text{lex}} x_{(n+3)/2}\ldots x_nx_1 \quad n \text{ odd} \\ &\vdots \\ x_1x_2\ldots x_{n-1} &\leq_{\text{lex}} x_nx_1\ldots x_{n-2} \end{aligned}$$

Proof We show that $\mathcal{M}$ is equivalent to the lex-leader constraints for C_n, namely

$$x_1\ldots x_n \leq_{\text{lex}} x_{k+1}\ldots x_nx_1\ldots x_k \tag{1}$$

for $1 \leq k < n$.

We will refer to the ith constraint in $\mathcal{M}$ as t_i. To see that the lex-leader constraints entail $\mathcal{M}$, note that each constraint in $\mathcal{M}$ is an initial subsequence of one of the lex-leader constraints.

We now prove the converse, we wish to show that $\mathcal{M}$ implies (1) for $1 \leq k < n$. The constraint t_k is $x_1\ldots x_k \leq_{\text{lex}} x_{k+1}\ldots x_m$, where $m = 2k$ if $k \leq n/2$ and $m = 2k - n$ otherwise. So the initial subsequence of length k of (1) holds, and we may assume without loss of generality that $x_1 = x_{k+1}, x_2 = x_{k+2}, \ldots, x_k = x_m$.

We need to prove that under these assumptions

$$x_{k+1}\ldots x_n \leq_{\text{lex}} x_{2k+1}\ldots x_nx_1\ldots x_k. \tag{2}$$

We have $x_1\ldots x_k = x_{k+1}\ldots x_m$, so we can rewrite the left-hand side of (2) as $x_1\ldots x_kx_{m+1}\ldots x_n$. The constraint t_{2k} is $x_1\ldots x_{2k} \leq_{\text{lex}} x_{2k+1}\ldots x_p$ where $p = (4k - 1 \bmod n) + 1$. Thus the initial subsequence of length k of (2) holds, and we may assume without loss of generality that $k < n/2$, and that $x_1 = x_{k+1} = x_{2k+1}$, $x_2 = x_{k+2} = x_{2k+2}, \ldots, x_k = x_{2k} = x_{3k}$.

If n is divisible by k then by induction (1) holds.

Assume that n is not divisible by k, let $b = \lfloor n/k \rfloor$ and $c = n \bmod k$ so that $n = bk + c$. We must show that

$$x_{bk+1}\ldots x_n \leq_{\text{lex}} x_{(b+1)k-n+1}\ldots x_k. \tag{3}$$

Note that (3) has length $c < k$, and that $(b+1)k - n + 1 = bk + k - (bk + c) + 1 = k - c + 1$.

Consider the constraint t_{k-c}, namely

$$x_1\ldots x_{k-c} \leq_{\text{lex}} x_{k-c+1}\ldots x_{2k-2c},$$

and recall that by assumption $x_1\ldots x_c = x_{bk+1}\ldots x_{bk+c}$, with $bk + c = n$. If $k - c \geq c$ then the first c pairs of variables in t_{k-c} are precisely what we need to prove.

Therefore, we assume that $k - c < c$, and show that under the additional assumption $x_{bk+1}\ldots x_{(b+1)k-c} = x_{k-c+1}\ldots x_{2k-2c}$ that

$$x_{(b+1)k-c+1}\ldots x_c \leq_{\text{lex}} x_{2k-2c+1}\ldots x_k.$$

However our initial assumption that $x_i = x_{k+i}$ for $1 \le i \le k$ entails $x_{(b+1)k-n+i} = x_i$ for $2k - 2c + 1 \le i \le k$, hence constraint (1) is entailed for $1 \le k < n$. □

Before the next lemma we consider what happens to the constraint t_{12} for the cyclic group C_{15} when we assume equality in the first 11 pairs of variables.

$$t_{12} : x_1x_2x_3\ x_4x_5x_6\ x_7x_8x_9\ x_{10}x_{11}x_{12} \le_{\text{lex}} x_{13}x_{14}x_{15}\ x_1x_2x_3\ x_4x_5x_6\ x_7x_8x_9$$

We assume that $x_1 = x_{13}$, that $x_2 = x_{14}$ and that $x_3 = x_{15}$. Then the most significant variables appear on the right hand side of the constraint, producing equality classes $\{x_1, x_{13}, x_4\}$, $\{x_2, x_{14}, x_5\}$ and $\{x_3, x_{15}, x_6\}$. Next the variables x_4, x_5 and x_6 appear on the right hand side of the constraint, enlarging the equality classes to $\{x_1, x_{13}, x_4, x_7\}$, $\{x_2, x_{14}, x_5, x_8\}$, $\{x_3, x_{15}, x_6, x_9\}$. The final two assumptions enlarge the first two equality classes to $\{x_1, x_{13}, x_4, x_7, x_{10}\}$ and $\{x_2, x_{14}, x_5, x_8, x_{11}\}$.

This pattern is explored further in the following lemma.

Lemma 7 *If $k > n/2$ then the assumption of equality in the first $(k-1)$ pairs of variables in t_k produces $n-k$ distinct equality classes of size greater than 1 between the variables involved in that constraint. If $k \le n/2$ then $(k-1)$ equality classes of size greater than 1 are produced.*

Proof We first consider the case that $k \le n/2$. Then t_k is $x_1 \dots x_k \le_{\text{lex}} x_{k+1} \dots x_{2k}$. All variables in t_k are distinct, hence $k-1$ equality classes of size greater than 1 are created.

For the rest of the proof we assume that $k > n/2$, so t_k is

$$x_1 \dots x_k \le_{\text{lex}} x_{k+1} \dots x_n x_1 \dots x_{2k-n}.$$

The $n-k$ most significant pairs in t_k contain distinct variables.

We assume that $x_1 = x_{k+1}$, $x_2 = x_{k+2}$, ..., $x_{n-k} = x_n$. After these $n-k$ pairs of variables there is a repeat of $x_1x_2x_3 \dots x_{(n-k)}$, this time on the right hand side. We assume that $x_1x_2x_3 \dots x_{n-k} = x_{(n-k)+1}x_{(n-k)+2} \dots x_{2(n-k)}$.

We now have $n-k$ equality classes, each involving 3 variables, e.g. $x_1 = x_{k+1} = x_{(n-k)+1}$. The pattern then continues with $x_{(n-k)+1}x_{(n-k)+2} \dots x_{2(n-k)}$ appearing on the right hand side, assumed to be equal to the next $n-k$ variables. This pattern continues until the pair of variables under consideration, namely $x_k \le x_{2k-n}$.

As the number of variables in each equality class grows, the additions are variables that have not appeared before. Therefore the initial $n-k$ equality classes will never merge.

The new variables are always assumed to be equal to a variable currently in an equality class of size greater than 1, so there will never be more than $n-k$ equality classes of size greater than 1. □

Theorem 5 *The reduced set of lex ordering constraints for the cyclic group is minimal.*

Proof We consider Rules 2 and 3' in turn, showing that each constraint can be reduced no further.

Rule 2: We start by showing that no further application of Rule 2 is possible by examining an arbitrary constraint t_k. The pair under consideration for removal from t_k by Rule 2 is: $x_k \leq_{\text{lex}} x_l$, where $l = 2k$ if $2k \leq n$ and $l = 2k - n$ if $2k > n$.

First note that if $2k \leq n$ then we have equality classes $x_1 = x_{k+1}, x_2 = x_{k+2}, \ldots, x_{k-1} = x_{2k-1}$. The constraints t_i with $i < k$ have most significant pairs $x_1 \leq x_2$, $x_1 \leq x_3, \ldots x_1 \leq x_{k-1}$. The constraints t_i with $i > k$ have most significant pairs $x_1 \leq x_{k+2}, x_1 \leq x_{k+3}, \ldots, x_1 \leq x_{n-1}$. In order to use less significant pairs of variables in these constraints we must show equality in these initial pairs, however they all lie in distinct equality classes. Thus we never assume x_k to be less than or equal to anything, and t_k cannot be reduced further.

We now assume that $2k > n$, so that the pair under consideration for removal from t_k by Rule 2 is $x_k \leq x_l$, where $l = 2k - n$.

By Lemma 7, the first $n - k$ decision variables are never assumed to be equal to each other. Notice also that these $n - k$ variables are equal to the last $n - k$ variables respectively, i.e., $x_1 = x_{k+1}, x_2 = x_{k+2}, \ldots, x_{n-k} = x_n$.

The equality classes not containing x_1 contain the variables $x_{k+2}, \ldots, x_n$. These are the right hand variables of the most significant pairs of t_i for $i > k$. Therefore the Rule 2 assumptions of equality alone are not enough to deduce equality in these most significant pairs and as such we cannot use less significant pairs to justify the removal of $x_k \leq x_l$.

We now consider the lower arity constraints. We have equality in the most significant pairs of $t_{n-k}, t_{2(n-k)}, \ldots$. This is simply because the left hand element in the pair is always x_1, and Lemma 7 shows us that x_1 is equal to every $(n-k)^{th}$ element. In these constraints, we find that the pairs of variables are matched to those in the equality classes that we are assuming exist from t_k. This is because the equality relations step along in groups of $n - k$.

The first distinct pair of variables in each constraint which are not assumed to be equal is the pair with x_k on its right hand side, hence the most we may deduce is that x_k is greater than another variable. Since x_k has not been assumed to be equal to anything else and since nothing we have done so far has entailed its equality to anything else, we cannot deduce that $x_k \leq x_l$.

Rule 3': We now consider application of Rule 3' on pairs of variables which are not least significant in their respective constraints. First we show that, as with application of Rule 2 on the least significant pairs, the assumptions and entailed equalities are insufficient to show equality in the most significant pairs of constraints of a larger arity.

The set of assumed equalities in application of Rule 3' on a pair of variables which are not least significant is a subset of the set of assumed equalities in the application of Rule 2 on the same constraint. As such, by Corollary 2, the most significant pairs in constraints t_i for $i > k$ are never assumed to be equal.

We now show that constraints t_i for $i < k$ do not entail equality in $x_i \leq x_j$. where $1 \leq i \leq (k-1)$, and $j = i + k$ if $k \leq n/2$ or $j = i - (n-k)$ if $k > (n/2)$.

Since all variables on any one side of a lex constraint are distinct, and since in the reduced set of cyclic lex constraints any variable occurs on the left hand side of the constraint before it occurs on the right hand side, the variable x_i is never assumed to be equal to any other variable in the application of Rule 3'. By Lemma 4, to show $x_i = x_j$ we require x_i to appear on the left hand side of another constraint.

a) Circular Golomb Ruler Problem: Given a circle with circumference n, place m ticks at integer points around the circle such that all inter-tick distances along the circumference are distinct. (n and m are both positive integers).

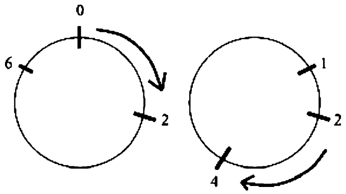
b)


Fig. 3 Specification of the Circular Golomb Ruler problem. Symmetric solutions to the length 7, 3-tick Circular Golomb Ruler problem

The variable x_i appears on the left hand side of constraints $t_i, t_{i+1}, \dots t_{k-1}$. These constraints have most significant pairs $x_1 \leq x_{i+1}, x_1 \leq x_{i+2}, \dots, x_1 \leq x_k$. We assume equalities between certain pairs of the variables $x_1, \dots, x_{i-1}$ and $x_{k+1}, \dots, x_n$, but no equalities in the most significant pairs of the constraints $t_i, t_{i+1}, \dots t_{k-1}$.

Examining Lemma 4, we require one of $x_{i+1}, x_{i+2}, \dots, x_k$ to appear on the left hand side of another pair to be able to deduce equality. The only constraints t_l for $l < k$, that have these variables on the left hand side are those where we are currently trying to deduce equality in the most significant pairs. As such we cannot consider any pairs, other than the most significant, in any of the constraints $t_i, t_{i+1}, \dots t_{k-1}$.

There is no support for the removal of $x_i \leq x_j$ by Rule 3', so the reduced set of lex constraints for the cyclic group is minimal. □

Figure 3a defines the Circular (or Modular) Golomb Ruler problem. Two solutions to the instance of this problem where n is 7 and m is 3 are shown in Fig. 3b. Clearly, these solutions are symmetric: one can be obtained from the other via rotation. Part of the symmetry group in the above problem, $n = 7$ and $m = 3$, is C_3. The symmetry breaking constraints required to order the set of ticks, $T = \{t_1, t_2, t_3\}$ are $t_1 \leq t_2$, and $t_1 t_2 \leq_{\text{lex}} t_3 t_1$.

6.3 Alternating groups

Any permutation can be written as a product of cycles of length 2, called *transpositions*. There is usually more than one way of writing any given permutation as a product of transpositions, but the parity of the number of transpositions occurring in all such products is fixed. The *alternating group* A_n on n points is the subgroup of the symmetric group S_n that contains all of the permutations that can be written as a product of an even number of transpositions. It contains exactly half of the permutations of the symmetric group, and hence could be expected to have a similarly small set of minimal lex constraints.

Theorem 6 *Let P be a CSP with n decision variables,* $\{x_1, x_2, ..., x_n\}$. *If the symmetry group of P is* A_n *on variables then a complete set C of symmetry breaking constraints is:*

$$\begin{aligned} x_{n-2} &\leq x_{n-1} & (c_1) \\ x_{n-2}x_{n-1} &\leq_{\text{lex}} x_n x_{n-2} & (c_2) \\ x_i x_{n-1} &\leq_{\text{lex}} x_{i+1} x_n & (c_{3,i}),\ 1 \leq i \leq n-3 \end{aligned}$$

Proof We first show that C is entailed by lex-leader constraints. The permutation $(n-2, n-1, n) \in \mathrm{A}_n$ entails c_1 and the permutation $(n-2, n, n-1)$ entails c_2. Finally, we have $(x_i, x_{i+1})(x_{n-1}, x_n) \in \mathrm{A}_n$ for $1 \leq i \leq n-3$, yielding $c_{3,i}$.

We now consider the converse and show that C entails the full set of lex-leader constraints, of size $(n!/2) - 1$. Consider c_1 and the first pair of variables in $c_{3,i}$, we see that the first $n-1$ variables are sorted. The first pair of variables in c_2, namely $x_{n-2} \leq x_n$ implies that the variables $x_1, x_2, \ldots, x_{n-2}, x_n$ are also sorted. So the only possibly unbroken symmetries move x_{n-1} and x_n. Since the permutation swapping (x_{n-1}, x_n) is odd, for there to be any remaining symmetries either $x_{n-1} = x_{n-2}$, or $x_n = x_{n-2}$, or some of the other variables have equal values.

If $x_{n-1} = x_n = x_{n-2}$ then there are no further symmetries to break, since $x_1 \ldots x_n$ is now sorted.

If exactly one of x_{n-1} and x_n is equal to x_{n-2}, then the other one is greater. If $x_{n-2} = x_n < x_{n-1}$ then we are violating c_2. Therefore $x_{n-2} = x_{n-1}$ so the solution is already minimal.

If $x_{n-1}, x_n \neq x_{n-2}$ then both x_{n-1} and x_n are strictly greater than x_{n-2} so the biggest two values occur at the end of the list of variables. This will be lex minimal under the alternating group unless $x_{n-1} > x_n$ and there are two equal values (say x_i, x_{i+1}) somewhere else in the full assignment. However in this instance we would violate the constraint $c_{3,i}$. □

Theorem 7 *The set C is minimal.*

Proof Application of Rule 2 for the removal of c_1 assumes $x_{n-2} \leq x_n$ and $x_i \leq x_{i+1}$ for $1 \leq i \leq n-3$. Hence, $x_1 \leq x_2 \leq \ldots \leq x_{n-2} \leq x_n$. The constraint c_1 is not implied and therefore cannot be removed by Rule 2.

To reduce c_2 by Rule 2 we assume that $x_{n-2} = x_n, x_{n-2} \leq x_{n-1}$ and $x_i \leq x_{i+1}$ for $1 \leq i \leq n-3$. From the transitive closure of these (in)equalities, $x_1 \leq x_2 \leq \ldots \leq x_{n-2}$ and $x_n = x_{n-2} \leq x_{n-1}$. The inequality $x_{n-1} \leq x_{n-2}$ is not entailed and cannot be removed by Rule 2.

To reduce $c_{3,j}$ by Rule 2 we assume that $x_j = x_{j+1}$, that $x_i \leq x_{i+1}$ for $1 \leq i \leq n-3$ with $i \neq j$, that $x_{n-2} \leq x_{n-1}$, and that $x_{n-2} \leq x_n$. From the transitive closure of these relations, $x_1 \leq x_2 \leq \ldots \leq x_j$, that $x_{j+1} \leq x_{j+2} \leq \ldots \leq x_{n-2} \leq x_{n-1}$ and that $x_{n-2} \leq x_n$. The inequality $x_{n-1} \leq x_n$ is not entailed and cannot be removed by Rule 2.

Consider now the action of Rule 3' on C. Rule 3' considers all but the least significant pairs in any constraint for removal, and so is not applicable to c_1 and c_2. To apply Rule 3' to $x_i \leq x_{i+1}$ in $c_{3,i}$ with $1 \leq i \leq n-3$, we assume $x_{n-2} \leq x_{n-1}$, $x_{n-2} \leq x_n$, and that $x_j \leq x_{j+1}$ for $1 \leq j \leq n-3$ with $j \neq i$. By Lemma 5, we require x_i to appear on the left hand side of another constraint, which does not occur. □

6.4 Dihedral groups

The *dihedral group* D_n is the symmetries of a regular n-sided shape, for example D_4 is the symmetry group of the square. Let

$$a = (1, 2, \ldots, n)$$

and

$$b = (1, n)(2, n-1) \ldots (\lfloor n/2 \rfloor, \lceil (n+3)/2 \rceil).$$

All elements of D_n have a unique decomposition as a product of the form $a^i b^j$ for $0 \le i \le n-1$ and $0 \le j \le 1$. The set of elements for which $j = 0$ is the cyclic group of order n, whereas the elements $a^i b$ all have order 2.

We define a set of $2n - 5$ symmetry breaking constraints as follows. First, the constraints t_i from the cyclic group for $2 \le i \le n-3$:

$$\begin{aligned}
x_1 x_2 &\le_{\text{lex}} x_3 x_4 \\
x_1 x_2 x_3 &\le_{\text{lex}} x_4 x_5 x_6 \\
&\vdots \\
x_1 x_2 \dots x_{n/2+1} &\le_{\text{lex}} x_{n/2+2} \dots x_n x_1 x_2 \quad n \text{ even} \\
x_1 x_2 \dots x_{(n-1)/2} x_{(n+1)/2} &\le_{\text{lex}} x_{(n+3)/2} \dots x_n x_1 \quad n \text{ odd} \\
&\vdots \\
x_1 x_2 \dots x_{n-3} &\le_{\text{lex}} x_{n-2} x_1 \dots x_{n-6}
\end{aligned}$$

Then a further $n-1$ constraints γ_i and δ_j. If $n = 2k+1$ is odd we define $s = 2$, $t = 1$ and $u = 0$, otherwise $n = 2k$ is even, and we let $s = 1$, $t = 0$ and $u = 1$. Then define

$$\begin{aligned}
\gamma_i :& \quad x_1 \dots x_i x_{2i+1} \dots x_{k+i} \le_{\text{lex}} x_{2i} \dots x_{i+1} x_n \dots x_{k+i+s} && 1 \le i \le k-u \\
\delta_j :& \; x_1 \dots x_j x_{2j+2} \dots x_{k+j+t} \le_{\text{lex}} x_{2j+1} \dots x_{j+2} x_n \dots x_{k+j+2} && 0 \le j \le k-1.
\end{aligned}$$

Note that the substrings of variables on the left hand side of γ_i and δ_j occur in increasing order, whilst those on the right hand side occur in decreasing order: if the subscripts at each end of the substring are decreasing on the left hand side, or increasing on the right hand side, then the substring is empty.

Theorem 8 *Let P be a CSP with n decision variables,* $\{x_1, x_2, \dots, x_n\}$. *If the symmetry group of P is the dihedral group* D_n *on variables then the set* $\{t_i, \gamma_j, \delta_l \; : 2 \le i \le n-3, 1 \le j \le k-u, \; 0 \le l \le k-1\}$ *of symmetry-breaking constraints is complete.*

Proof The constraints can be divided into two sets. Those derived from the cyclic group symmetry over the same decision variables, namely the t_is, and those derived from the remainder of the dihedral group permutations, namely the γ_is and δ_js.

The constraints t_i *for* $2 \le i \le n-3$. We have three fewer constraints than for C_n. These three constraints break the symmetry represented by the permutations a, a^{n-2} and a^{n-1}, and would correspond to constraints t_1, t_{n-2} and t_{n-1}. We show that these missing constraints are entailed by the remaining constraints.

The constraint corresponding to the permutation a is $x_1 \le x_2$, which is entailed by γ_1.

The constraint corresponding to the permutation a^{n-1} is

$$x_1 \dots x_{n-1} \le_{\text{lex}} x_n x_1 \dots x_{n-2}.$$

We show that t_{n-1} is entailed by the other constraints. Consider the first pair of variables, $x_1 \le x_n$. Constraint δ_0 has most significant pair $x_2 \le x_n$, and t_1 requires $x_1 \le x_2$, which together entail $x_1 \le x_n$.

Assuming equality in the first z pairs of variables of t_{n-1} forces

$$x_1 = x_n = x_2 = \dots = x_z \quad \text{where } 2 \le z \le n-2.$$

The next pair under consideration is $x_{z+1} \leq x_z$.

First assume that $z = 2i \leq n-2$ is even, then constraint γ_i is

$$x_1 \dots x_i x_{2i+1} \dots x_{k+i} \leq_{\text{lex}} x_{2i} x_{2i-1} \dots x_{i+1} x_n \dots x_{k+i+s},$$

where $s = 2$ or 1 according as n is odd or even. The first i pairs of variables in this constraint have been assumed to be equal, so we deduce that $x_{2i+1} = x_{z+1} \leq x_n = x_z$, as required.

Next assume that $z = 2i + 1 \leq n - 2$ is odd, then constraint δ_i is

$$x_1 \dots x_i x_{2i+2} \dots x_{k+i+t} \leq_{\text{lex}} x_{2i+1} x_{2i} \dots x_{i+2} x_n \dots x_{k+i+2},$$

where t is 0 or 1 according as n is even or odd. Since the first z pairs of variables in this constraint have been assumed to be equal, we deduce that $x_{z+1} = x_{2i+2} \leq x_n = x_z$, as required. Hence, t_{n-1}.

Finally we consider the constraint corresponding to a^{n-2}, namely t_{n-2} which is

$$x_1 x_2 \dots x_{n-2} \leq_{\text{lex}} x_{n-1} x_n x_1 \dots x_{n-4}.$$

The inequality $x_1 \leq x_{n-1}$ is the most significant pair in δ_{k-1} (even values of n) or γ_k (odd values of n). Also, $x_2 \leq x_n$ is the most significant pair in δ_0. If we assume that $x_1 = x_{n-1}$ and $x_2 = x_n$, then δ_0 gives $x_3 \leq x_{n-1} = x_1$, as required.

Let us now consider the general case $x_{z+1} \leq x_{z-1}$ for $3 \leq z \leq n-2$, so that we assume that $x_1 = x_n = x_3 = x_5 = \cdots$ and $x_2 = x_{n-1} = x_4 = \cdots$, with the highest subscript (other than n or $n-1$) in an equality class of size greater than 1 being x_z. If $z = 2i + 1$ is odd then the constraint δ_i has the first n pairs of variables equal, so we deduce that $x_{2i+1} = x_{z+1} \leq x_n = x_{z-1}$. If $z = 2i$ is even then the constraint δ_{i-1} has first $i - 1$ pairs of variables equal. We then find $x_{2i} = x_z = x_n$, so continuing to the next variable we deduce $x_{2i+1} = x_{z+1} \leq x_{n-1} = x_{z-1}$, as required.

Since the missing constraints from the cyclic set, which themselves are known to be complete from Theorem 5, are entailed by the additional constraints to break the dihedral symmetry, the reduced set of cyclic constraints are complete with respect to the cyclic symmetry in the dihedral group.

The remaining constraints Next we show that the reduced set of lex-constraints entail all constraints corresponding to the remaining elements of the dihedral group, namely those of the form $a^i b$. Recall that the elements of the form $a^i b$ have order 2, and so are a product of disjoint 2-cycles. The second variable of each 2-cycle will be deleted from each constraint by Rule 1. Also, when n is even, half of the permutations $a^i b$ fix two variables, which will also be deleted by Rule 1: these correspond to δ_i for $0 \leq i \leq k - 1$.

The full set of lex constraints has n symmetry breaking constraints for the remaining dihedral permutations, but that the reduced set only has $n - 1$. The constraint which breaks the symmetry

$$b = (1, n)(2, n-1) \dots (\lfloor n/2 \rfloor, \lceil (n+2)/2 \rceil)$$

has been completely removed. We now show that this constraint is entailed by the reduced set of constraints. The unreduced form of this missing constraint c is

$$x_1 \dots x_n \leq_{\text{lex}} x_n \dots x_1.$$

We consider $t_n : x_1 \dots x_n \leq_{\text{lex}} x_n x_1 \dots x_{n-2}$, which has most significant pair $x_1 \leq x_n$. These are the first pair of variables in c.

Consider now an arbitrary pair $x_z \leq x_{n-z+1}$ for $2 \leq z \leq n/2$. When considering this pair for removal we assume that $x_1 = x_n$, $x_2 = x_{n-2}$, ..., $x_{z-1} = z_{n-z}$. Therefore from t_n and t_1 we deduce $x_2 = x_1 = x_n = x_{n-2}$. We then deduce from t_n and t_i that $x_i = x_1$ for $i \in \{1, \dots, n\}$. Hence the rest of c is entailed and can be removed.

We have shown that the reduced set of lex ordering constraints for the dihedral group entails the complete set. Since the converse is clear we conclude that the reduced set of dihedral lex constraints is sound and complete. □

Theorem 9 *The reduced set of dihedral group symmetry breaking constraints is minimal.*

Proof By inspection the most significant pair of every constraint in the reduced set contains two distinct variables, as such, From Lemma 6, we will only consider the applications of Rule 3′ where assumptions of equality are made.

The constraints can be divided into two sets. Those derived from the cyclic group symmetry over the same decision variables, namely the t_is, and those derived from the remainder of the dihedral group permutations, namely the γ_is and δ_js.

The constraints t_i for $2 \leq i \leq n-3$. We know from Theorem 5 that the constraints t_i, for $2 \leq i \leq n-3$, themselves cannot be reduced using only themselves as support. We therefore look to using the set of constraints generated by γ_i and δ_j, for $1 \leq i \leq k-u$ and $0 \leq j \leq k-1$.

First recall from Lemma 7 that the number of equivalence classes produced from assumptions of equality in the constraint t_i, for $2 \leq i \leq n-3$, is $n-k$ (or k, $k \leq n/2$).

Consider the pair, $x_b \leq x_c$, for removal by Rule 2 or Rule 3' to be in the constraint t_k, $2 \leq k \leq n/2$. Suppose that these assumed equalities allow us to deduce equality in the most significant pair, $x_1 \leq x_m$, in some γ_i or δ_j, for $2 \leq i \leq k-u$ and $2 \leq j \leq k-1$. The next most significant pair, $x_2 \leq x_{m-1}$, is then deduced from this constraint. Since $x_1 = x_m$, $x_{m-1} = x_{n-k}$ from Lemma 7. The equivalence classes are of size two, therefore x_2 is not assumed equal to x_{m-1}, as such we cannot consider less significant pairs in that constraint.

We now show that $x_2 \leq x_{m-1}$ itself cannot provide support for the removal of any pair in t_i, for $2 \leq i \leq n-3$. Notice that x_b is never assumed to be equal to any other variable under the actions of Rules 2 and 3′, as such for $x_2 \leq x_{m-1}$ to provide support for the removal of $x_b \leq x_c$, we require $b = 2$.

First consider the case where $x_b \leq x_c$ is the least significant pair, and as such we reduce using Rule 2. Observe that the only constraint where x_2 occurs on the left hand side of any least significant pair is in the constraint $x_1x_2 \leq x_3x_4$. Assuming $x_1 = x_3$ allows us to inspect the next most significant pair in only one constraint. This is the constraint beginning $x_1x_4 \leq_{\text{lex}} x_3x_n$. Obviously this provides no support for the removal of $x_2 \leq x_4$.

Next we consider the case where $x_b \leq x_c$ is not the least significant pair, and as such we reduce using Rule 3′. Since x_b is not assumed to equal anything else can only consider pairs with x_2 on the left hand side. If the pair under consideration is

$x_2 \le x_{m+1}$, then $x_1 = x_m$, and $x_2 \le x_{m-1}$ is entailed by some constraint γ_i or δ_j, for $2 \le i \le k-u$. Since x_{m-1} is not assumed equal to anything else, $x_2 \le x_{m-1}$ cannot provide support for the removal of $x_2 \le x_{m+1}$.

Now consider the constraint t_k, $n/2 < k \le n$ and the pair, $x_b \le x_c$, for removal by Rule 2 or Rule 3′. Suppose that these assumed equalities allow us to deduce equality in the most significant pair, $x_1 \le x_m$, in some γ_i or δ_j, for $2 \le i \le k-u$ and $2 \le j \le k-1$. The next most significant pair, $x_2 \le x_{m-1}$, is then entailed by this constraint. We first show that x_2 is not entailed equal to x_{m-1}. Since $x_1 = x_m$, $x_{m-1} = x_{n-k}$ from Lemma 7. The minimum value of $n-k$ is 3, therefore x_2 is not entailed equal to x_{m-1}, as such we cannot consider other less significant pairs in that constraint.

We now show that $x_2 \le x_{m-1}$ itself cannot provide support for the removal of any pair in t_i, for $2 \le i \le n-3$. Observe that x_b is never assumed to be equal to any other variable under the actions of Rules 2 and 3′, as such for $x_2 \le x_{m-1}$ to provide support for the removal of $x_b \le x_c$, we require $b = 2$.

Notice that in this subset of the constraints $x_b \le x_c$ is never the least significant pair, and as such we reduce using Rule 3′. Since x_b is not assumed to equal anything else we can only consider pairs with x_2 on the left hand side. If the pair under consideration is $x_2 \le x_{m+1}$, then $x_1 = x_m$, and $x_2 \le x_{m-1}$ is entailed by some constraint γ_i or δ_j, for $2 \le i \le k-u$. Since x_{m-1} is not assumed equal to anything else, $x_2 \le x_{m-1}$ cannot provide support for the removal of $x_2 \le x_{m+1}$.

As special cases consider support from the constraints generated by δ_0, δ_1,and γ_1.

δ_0 produces the constraint with most significant pair $x_2 \le x_n$. We know that x_n is assumed to be equal to x_{n-k}. Since the minimum value of n-k is 3, x_2 is never assumed to be equal to x_n.

δ_1 produces the constraint with most significant pair $x_1 \le x_3$. The only time x_1 is assumed to equal x_3 is in the case described above, in all other cases, since $n-k$ is at least 3, this is not the case.

γ_0 has most significant pair $x_1 \le x_2$. Again, since $n-k$ is at least 3, x_1 is never assumed to be equal to x_2.

The remaining constraints We now consider reducing the remaining constraints. First notice that every variable in any one of these constraints is distinct. Considering the pair $x_b \le x_c$ for removal in the constraint with most significant pair $x_1 \le x_m$ allows us to assume $x_1 = x_m$. Notice also that every other constraint, with the exception of δ_0 which we will consider as a special case, has x_1 on the left hand side of its most significant pair.

Assuming $x_1 = x_m$ allows the inspection of the next most significant pair in at most one other constraint. The next most significant pair in this constraint is $x_2 \le x_{m+1}$. We now consider two cases.

Consider the case where $b \ne 2$. Here we assume that either $x_2 = x_{m-1}$ or we don't assume anything about x_2. In both examples x_2 is not assumed equal to x_{m+1} and as such we cannot deduce inequalities from less significant pairs in that constraint. Note also that since all variables in the constraint under consideration are distinct that the pair $x_2 \le x_{m+1}$ cannot provide support for the removal of $x_b \le x_c$.

Now consider the case where $b = 2$. The pair under consideration in this case is $x_2 \le x_{m-1}$, with the exception of γ_1 which we consider as a special case. Here we assume nothing about x_2. Since all variables in the constraint under consideration are distinct the pair $x_2 \le x_{m+1}$ cannot provide support for the removal of $x_2 \le x_{m-1}$.

We can only inspect the most significant pair in γ_1 since the most significant pair is $x_1 \le x_2$, and x_1 is never assumed to be equal to x_2.

Where δ_0 is the constraint under consideration we make no assumptions about x_1. Since all other constraints have x_1 in their most significant pair we can consider no less significant pairs. □

7 Combining groups

Complicated symmetry groups can often be built up out of smaller, easier-to-describe groups. When this occurs we say that the symmetry group is the *product* of the smaller groups. In this section we analyse two of the most commonly-occurring group products that occur in constraint satisfaction problems, and find sets of constraints for the products in terms of sets of constraints for the smaller groups.

7.1 Direct products

One of the most common situations is that the variables of a CSP can be partitioned into two disjoint sets, where the symmetries of the CSP act independently on each set. In this situation we have a *direct product*.

By $\mathrm{Sym}(\Omega)$ we denote the symmetric group on a set Ω, i.e., the set of all bijections from Ω to itself.

Definition 4 Let $G \le \mathrm{Sym}(\Omega)$ and $H \le \mathrm{Sym}(\Delta)$ be groups, with Ω and Δ disjoint sets. The *direct product* of G and H, written $G \times H$, is the set $\{(g, h) : g \in G, h \in H\}$, with coordinatewise multiplication. Elements of $G \times H$ permute the set $\Omega \cup \Delta$ as follows: $(g, h)(x) = g(x)$ if $x \in \Omega$, and $(g, h)(x) = h(x)$ if $x \in \Delta$.

Two important subgroups of $G \times H$ are the set $\{(g, 1_H) : g \in G\}$ and the set $\{(1_G, h) : h \in H\}$, where 1_G denotes the identity element of G and 1_H the identity element of H.

Consider now the problem of finding two circular Golomb rulers, as in Fig. 3. Part of the symmetry in this problem is the direct product of the two cyclic groups of symmetries for each ruler, as we can act on each ruler independently of the other one: later in this section we'll consider further symmetries.

For this small example take two sets of ticks, $T = \{t_1, t_2, t_3\}$ and $T' = \{t'_1, t'_2, t'_3\}$, defining two circular golomb rulers for $n = 7$ $m = 3$. We can see that the lex constraints required to break the two sets of cyclic symmetries are $t_1 \le t_2$, $t'_1 \le t'_2$, $t_1t_2 \le_{\mathrm{lex}} t_3t_1$ and $t'_1t'_2 \le_{\mathrm{lex}} t'_3t'_1$. Notice that these constraints are just the union of the constraints on T and those on T'. Our next theorem shows that this observation generalises.

In the following theorem, the variables of a CSP P are partitioned into two sets $\chi_1 = \{x_1, \ldots, x_n\}$ and $\chi_2 = \{y_1, \ldots, y_m\}$. For convenience, we assume that the chosen variable ordering for $\chi_1 \cup \chi_2$ induces the same ordering $x_1x_2 \ldots x_n$ on χ_1 as was used to write down symmetry breaking constraints for χ_1. We also assume that the chosen variable ordering induces the original ordering on χ_2, although the x_is and y_js may be interleaved. If this is not the case, then consistently replacing x_i in the constraints for

χ_1 by the new ith variable of χ_1, and similarly for y_j in χ_2, will produce the required constraints.

Theorem 10 *Let P be a CSP whose decision variables are partitioned into two disjoint sets* $\chi_1 = \{x_1, \ldots, x_n\}$ *and* $\chi_2 = \{y_1, \ldots, y_m\}$. *Assume that all symmetries of P are variable symmetries, and that the symmetries of P act independently on* χ_1 *and* χ_2, *with groups G and H of variable symmetries respectively. Let* L_G *be a minimal set of complete symmetry breaking constraints for G, and let* L_H *be a minimal set of complete symmetry breaking constraints for H. Then the symmetry group of P is* $G \times H$, *and a minimal complete set of symmetry breaking constraints for P is* $L_G \cup L_H$.

Proof The claim that the symmetry group of P is $G \times H$ is immediate.

We must show three things. Firstly, we show that the lex-leader constraints for $G \times H$ entail $L_G \cup L_H$, secondly that $L_G \cup L_H$ entails all of the lex-leader constraints, and finally that $L_G \cup L_H$ is minimal.

The lex-leader constraints for $G \times H$ will include a constraint c_g for each element of $G \times H$ of the form $(g, 1_H)$ where $g \in G$. Since c_g has all variables y_i in the same positions on each side, by Rule 1 c_g can be reduced to a constraint involving only the x_is, and hence the set of all such reduced c_gs entails all constraints in L_G. Similarly, the constraints for $G \times H$ will include a constraint c_h for each element $(1_G, h) \in G \times H$, and the constraint c_h can be reduced by Rule 1 to a constraint involving only the y_is. Hence all constraints in L_H are entailed by the lex-leader constraints.

Now we must show the converse. Let $a := (g, h)$ in $G \times H$, let $k = m + n$ and let $z_i \in \chi_1 \cup \chi_2$ for $1 \leq i \leq k$. We define an action of a on $\{z_i : 1 \leq i \leq k\}$ by $z_{a(i)} = x_{g(j)}$ if $z_i = x_j$ and $z_{a(i)} = y_{h(j)}$ if $z_i = y_j$. Then any lex-leader constraint is of the form:

$$z_1 z_2 \ldots z_k \leq_{\text{lex}} z_{a(1)} z_{a(2)} \ldots z_{a(k)}.$$

Since by assumption L_G and L_H are complete sets of constraints for G and H, they entail the constraints

$$x_1 \ldots x_n \leq_{\text{lex}} x_{g(1)} \ldots x_{g(n)} \ (A) \quad y_1 \ldots y_m \leq_{\text{lex}} y_{h(1)} \ldots y_{h(m)} \ (B).$$

Suppose without loss of generality that $z_1 = x_1$. Then from constraint (A) $z_1 \leq_{\text{lex}} z_{a(1)}$, so the first pair of the lex-leader constraint is entailed by $L_G \cup L_H$.

Suppose now that the first i pairs of the lex-leader constraint have been assumed to be equal, that is $z_1 = z_{a(1)}, z_2 = z_{a(2)}, \ldots, z_i = z_{a(i)}$. We show that together with $L_G \cup L_H$ this entails $z_{i+1} \leq z_{a(i+1)}$ and hence by induction the full lex-leader constraint. We have $z_{i+1} = x_j$ or $z_{i+1} = y_j$, let us assume without loss of generality that $z_{i+1} = x_j$ for some j. Then we must already have assumed that $x_1 = x_{g(1)}, x_2 = x_{g(2)}, \ldots, x_{j-1} = x_{g(j-1)}$, so (A) entails $z_{i+1} = x_j \leq x_{g(j)} = z_{a(i+1)}$, as required.

We finish by showing that $L_G \cup L_H$ is minimal. This follows from the fact that each constraint in $L_G \cup L_H$ involves only variables from χ_1 or only variables from χ_2. Hence constraints from L_G do not entail any additional equalities in constraints from L_H, and *vice versa*. Thus, since L_G and L_H were assumed to be minimal, the same holds for $L_G \cup L_H$. □

7.2 Wreath products

Another commonly arising way of combining two groups is the *imprimitive wreath product.*

Definition 5 Let $G \leq S_n$ and $H \leq S_k$. The *imprimitive wreath product* of G and H, denoted $G\mathrm{Wr}H$, is a subgroup of S_{nk}. It acts on k copies of the set of size n on which G acts. We have $G\mathrm{Wr}H = \{h(g_1, \ldots, g_k) : h \in H, g_i \in G\}$, and these elements permute the set $\{(i, j) : 1 \leq i \leq n, 1 \leq j \leq k\}$ by $h(g_1, \ldots, g_k)(i, j) = (g_j(i), h(j))$.

An example of such a group occurs in the social golfers problem. This problem asks one to partition a set of golfers into equal sized groups in each week of a tournament such that no golfer plays any other more than once. The symmetric group S_n interchanges each of the n golfers in each group, and the symmetric group S_k interchanges each of the k groups in a week, so $S_n\mathrm{Wr}S_k$ acts on the golfers in each week.

The full symmetry group of the two three-tick golomb rulers problem is $D_3\mathrm{Wr}S_2$. This is because the set of ticks on one ruler can be interchanged with the set of ticks on the other, and each set of ticks can be cyclically permuted or reversed. To break these symmetries we use the constraints:

$$t_1 \leq t_2 \qquad t'_1 \leq t'_2 \qquad t_2 \leq t_3 \qquad t'_2 \leq t'_3 \quad t_1 t_2 t_3 \leq_{\mathrm{lex}} t'_1 t'_2 t'_3$$

Notice that these constraints are equivalent to posting the dihedral group D_3 on each ruler separately, and then requiring the first ruler to be lexicographically less than or equal to the second. We now show that this observation generalises.

Suppose we have a set L_G of symmetry breaking constraints for a group G acting on variables $x_1, \ldots, x_n$, and a set L_H of symmetry breaking constraints for a group H acting on variables $y_1, \ldots, y_k$. Then $G\mathrm{Wr}H$ acts on a set of nk variables, x_{ij}, with $1 \leq i \leq n$ and $1 \leq j \leq k$. The group $G\mathrm{Wr}H$ has size $|G|^k \times |H|$, so writing down one constraint for each nontrivial group element is impractical. We now show how to reduce this to $k|L_G| + |L_H|$ constraints, assuming that our chosen variable ordering for the full CSP is

$$x_{11} \ldots x_{n1} x_{12} \ldots x_{n2} \ldots x_{nk}.$$

We first post $k|L_G|$ constraints, namely a copy of L_G on x_{ij} for each j. That is, we lex order each block of variables with respect to G. The arity of these constraints is the same as in L_G.

We then restate the constraints from L_H so that instead of being statements about $y_1, \ldots, y_k$, they are statements about the values of the second subscripts in sequences of strings of the form $x_{1j} x_{2j} \ldots x_{nj}$. For example, if we previously had a constraint $y_1 y_2 \leq y_2 y_3$ we would replace that with the constraint $x_{11} x_{21} \ldots x_{n1} x_{12} x_{22} \ldots x_{n2} \leq x_{12} x_{22} \ldots x_{n2} x_{13} x_{23} \ldots x_{n3}$. This results in $|L_H|$ constraints, each of arity n times their original arity.

Definition 6 Let L_G be a set of symmetry breaking constraints on $x_1, \ldots, x_n$, and let L_H be a set of symmetry breaking constraints on $y_1, \ldots, y_k$.

Then $L_G \mathrm{Wr} L_H$ is defined to be a set of symmetry breaking constraints on x_{ij} for $1 \le i \le n$ and $1 \le j \le k$. These constraints fall into two sets, of which the first set contains constraints of the form

$$x_{i_1 j} x_{i_2 j} \dots x_{i_s j} \le_{\text{lex}} x_{i'_1 j} x_{i'_2 j} \dots x_{i'_s j} \quad 1 \le j \le k$$

for all $x_{i_1} x_{i_2} \dots x_{i_s} \le_{\text{lex}} x_{i'_1} x_{i'_2} \dots x_{i'_s} \in L_G$. The second set contains constraints

$$x_{1 j_1} x_{2 j_1} \dots x_{n j_1} x_{1 j_2} \dots x_{n j_s} \le_{\text{lex}} x_{1 j'_1} x_{2 j'_1} \dots x_{n j'_1} x_{1 j'_2} \dots x_{n j'_s}$$

for all $y_{j_1} \dots y_{j_s} \le_{\text{lex}} y_{j'_1} \dots y_{j'_s} \in L_H$. If L_G and L_H are reduced from lex-leader constraints, so that $i' = g(i)$ for some permutation g of $\{1, \dots, n\}$ and $j' = h(j)$ for some permutation h of $\{1, \dots, k\}$, then we call constraints of the first type $c_{g,j}$ and of the second type c_h.

Theorem 11 *If L_G and L_H are complete sets of symmetry breaking constraints for groups G and H then $L_G \mathrm{Wr} L_H$ is a complete set of symmetry breaking constraints for $G \mathrm{Wr} H$ in the imprimitive action.*

Proof First we show that the constraints in $L_G \mathrm{Wr} L_H$ are entailed by the lex-leader constraints. The group $G \mathrm{Wr} H$ contains elements of the form

$$1_H(1_G, 1_G, \dots, 1_G, g, 1_G, \dots, 1_G)$$

for each $g \in G$, where g can occur in each coordinate. Using these group elements, applying Rule 1 to the resulting constraints, and then reasoning as in G, we produce each constraint $c_{g,i}$.

The group $G \mathrm{Wr} H$ also contains elements of the form $h(1_G, \dots, 1_G)$ for each $h \in H$. These elements produce all constraints of type c_h.

Next we must show that $L_G \mathrm{Wr} L_H$ entails all of the lex-leader constraints. An arbitrary element of $G \mathrm{Wr} H$ is of the form $a := h(g_1, \dots, g_k)$, and produces the constraint:

$$x_{11} \dots x_{n1} x_{12} \dots x_{nk} \le_{\text{lex}} x_{g_1(1)h(1)} \dots x_{g_1(n)h(1)} x_{g_2(1)h(2)} \dots x_{g_k(n)h(k)},$$

which we will denote by c_a. We must show that c_a is entailed by $L_G \mathrm{Wr} L_H$.

The constraint $y_1 \dots y_k \le_{\text{lex}} y_{h(1)} \dots y_{h(k)}$ is entailed by L_H, since L_H is assumed to be complete. Hence the constraints $L_G \mathrm{Wr} L_H$ entail

$$x_{11} \dots x_{n1} x_{12} \dots x_{nk} \le_{\text{lex}} x_{1h(1)} \dots x_{nh(1)} x_{1h(2)} \dots x_{nh(k)},$$

denoted α_h.

For $1 \le i \le k$ the constraints L_G entail

$$x_1 x_2 \dots x_n \le_{\text{lex}} x_{g_i(1)} x_{g_i(2)} \dots x_{g_i(n)},$$

as they are a complete set of symmetry breaking constraints for G. Hence for $1 \le i \le k$ and $1 \le j \le k$, the set $L_G \mathrm{Wr} L_h$ entails

$$x_{1j} x_{2j} \dots x_{nj} \le_{\text{lex}} x_{g_i(1)j} x_{g_i(2)j} \dots x_{g_i(n)j},$$

denoted $\beta_{g_i,j}$.

We will use these constraints to show that c_a is entailed by $L_G \mathrm{Wr} L_H$, considering the variables in blocks of n. Firstly, we have

$$x_{11}x_{21}\dots x_{n1} \leq_{\text{lex}} x_{1h(1)}\dots x_{nh(1)}$$

as the first n variable pairs from α_h. Considering $\beta_{g_1,h(1)}$ we also have

$$x_{1h(1)}\dots x_{nh(1)} \leq_{\text{lex}} x_{g_1(1)h(1)}\dots x_{g_1(n)h(1)}.$$

Combining these two inequalities we deduce that the first n pairs of variables of c_a are entailed by $L_G \mathrm{Wr} L_H$.

Suppose that we have shown that the first $n(i-1)$ variable pairs of c_a are entailed by $L_G \mathrm{Wr} L_H$, and that we are now considering pairs $n(i-1)+1,\dots,ni$, namely

$$x_{1i}x_{2i}\dots x_{ni} \leq_{\text{lex}} x_{g_i(1)h(i)}x_{g_i(2)h(i)}\dots x_{g_i(n)h(i)}.$$

To consider these variables we assume equality in the preceding $n(i-1)$ variable pairs, so $x_{11}=x_{1h(1)}=x_{g_1(1)h(1)}, x_{21}=x_{2h(1)}=x_{g_1(2)h(1)},\dots,x_{n(i-1)}=x_{nh(i-1)}=x_{g_{i-1}(n)h(i-1)}$. Considering constraint α_h we now deduce that

$$x_{1i}x_{2i}\dots x_{ni} \leq_{\text{lex}} x_{1h(i)}x_{2h(i)}\dots x_{nh(i)},$$

whereas from constraint $\beta_{g_i,h(i)}$ we deduce that

$$x_{1h(i)}x_{2h(i)}\dots x_{nh(i)} \leq_{\text{lex}} x_{g_i(1)h(i)}x_{g_i(2)h(i)}\dots x_{g_i(n)h(i)}$$

and so the result follows by induction. □

It is not necessarily the case that this construction produces minimal sets of constraints. For example, suppose that on a set $\{x_1, x_2\}$ of variables we have posted:

$$x_1 \leq x_2, \quad x_2 \leq x_1$$

and on a set $\{y_1, y_2\}$ of constraints we have posted $y_1 \leq y_2$. Both of these sets of constraints are minimal. Then for the wreath product we would post:

$$x_{11} \leq x_{21},\ x_{21} \leq x_{11},\ x_{12} \leq x_{22},\ x_{22} \leq x_{12},\ x_{11}x_{21} \leq_{\text{lex}} x_{12}x_{22}.$$

Consider the second pair of variables in the last constraint. To remove these, we assume that $x_{11}=x_{12}$, which considering the first four constraints yields $x_{21}=x_{11}=x_{12}=x_{22}$, so that $x_{21}=x_{22}$ and the final pair of variables may be deleted.

Note that the wreath product construction does generally produce minimal sets of constraints, for example when considering the wreath product of two symmetric groups the constraints require one to lex-order each block and then lex-order the blocks, which is clearly a minimal set of instructions.

We finish this section by noting that since both the direct product construction and the wreath product construction produce a number of constraints that is linear in the number of variables, whenever L_G and L_H are linear in their numbers of variables, our constructions may be iterated and will always produce a linear number of lexicographic constraints.

Fig. 4 Results from the testing of two circular golomb ruler instances

$n = 50$ $m = 6$	No Lex	Full Lex	Reduced Lex
Setup Time (s)	0	0.08	0.05
Memory (bytes)	728	1932	1872
Total Time (s)	5 Hours +	20.06	17.52
Nodes	unknown	344035	344035
Solutions	unknown	3600	3600
$n = 60$ $m = 7$	No Lex	Full Lex	Reduced Lex
Setup Time (s)	0	0.13	0.06
Memory (bytes)	976	2416	2356
Total Time (s)	5 Hours +	560.07	450.67
Nodes	unknown	6698526	6698526
Solutions	unknown	3578	3578

8 Experimental evaluation

Firstly we test a single instance circular golomb ruler problem. Figure 4 details results for the $n = 50$, $m = 6$ instance and then the $n = 60$, $m = 7$ instance. This complete symmetry group of the single instance circular golomb ruler is D_n. We therefore run one test using lex constraints for every permutation of the dihedral group, Full Lex, one test using the general formula for the dihedral group, Reduced Lex, and a benchmark test using no symmetry breaking constraints, No Lex. We found small but worthwhile savings on both time taken and memory required on a problem which is not solvable in a useful time frame without symmetry breaking constraints.

The next test uses the class scheduling problem: given d days, h hours per day and c classes, where $d \times h$ is divisible by c, find a timetable such that:

1. Each class has $(d \times h)/c$ assigned hours in the schedule.
2. No class has more than 2 hours in any given day.

Here the full symmetry group is the wreath product of two symmetric groups, $S_h \mathrm{Wr} S_d$. The first column shows the progress of a model with no symmetry breaking constraints. The next column shows the same model with the symmetry group broken by the full set of lex leader constraints; there are $(h!)^d \times (d!)$ such permutations. Finally, the last column shows the same model using the reduced wreath product symmetry breaking constraints; there are $(d \times (h - 1)) + (d - 1)$ such constraints. Figure 5 shows the results of two test instances of the class scheduling problem. Here the Reduced Lex constraints are more efficient in terms of memory used, time taken and nodes searched over. We attribute the additional nodes in this case to a feature of the solver used for testing.[2] It is worth noting that larger instances are not listed because the specification for the Full Lex in those cases was too large to compute. The specifications for both the Reduced Lex and the No Lex versions took a negligible amount of time to create.

In summary, for small groups the use of minimal sets of lexicographic ordering constraints over the lex leader method produces a small but significant improvement in efficiency. As one might expect, however, when we consider more complex groups

[2] The GACLex algorithm creates new copies of variables such that every variable in any one constraint is unique. This results in simple implications being missed, for example $(A \leq A) = \mathit{true}$.

Fig. 5 Results from the testing of two class scheduling instances

$d = 3\ h = 3\ c = 3$	No Lex	Full Lex	Reduced Lex
Setup Time (s)	0	0.02	0
Memory (bytes)	232	15784	328
Total Time (s)	0.046	0.2	0.031
Nodes	3304	119	45
Solutions	1512	6	6
$d = 4\ h = 3\ c = 6$	No Lex	Full Lex	Reduced Lex
Setup Time (s)	0	0.235	0
Memory (bytes)	376	373624	508
Total Time (s)	74.78	377.25	0.062
Nodes	22462011	57845	6318
Solutions	7484400	715	715

constructed by combining these groups this gain is magnified many times and the improvement obtained is substantial.

9 Conclusion and future work

This paper has concerned the use of sets of lexicographic ordering constraints for breaking variable symmetries in constraint satisfaction problems. Minimal (with respect to existing reduction rules) ordering constraints were derived for commonly-arising groups, such as the cyclic and dihedral groups. Furthermore, it was shown how to use these minimal sets as building blocks when considering common ways of combining groups, such as the direct product and imprimitive wreath product. Experimental results confirmed the value of minimal sets of ordering constraints, which can now be generated much more cheaply than with previous methods.

In future work, we will continue our investigation of individual and combined groups, and their corresponding sets of lexicographic ordering constraints. A particularly important example is the primitive wreath product, which corresponds to "row and column" symmetry [3], a very common feature of many constraint models. An important application of this work lies in automated constraint modelling: modelling decisions often introduce symmetry. By recognising this fact, we can build up a group-theoretic description of the symmetry in a model as the model is constructed. Our results can then be used to generate a minimal set of symmetry-breaking constraints for the model automatically. Hence, interfacing with an automated modelling system, such as Conjure [5], is an important item of future work.

Acknowledgements Andrew Grayland is supported by an EPSRC CASE studentship, sponsored by Microsoft Research. Ian Miguel is supported by a Royal Academy of Engineering/EPSRC Research Fellowship. Colva Roney-Dougal is partly funded by the Nuffield Foundation and partly by EPSRC grant number EP/C523229/1. We thank our anonymous reviewers and Youssef Hamadi for useful suggestions.

References

1. Cohen, D., Jeavons, P., Jefferson, C., Petrie, K.E., Smith, B.M.: Symmetry definitions for constraint satisfaction problems. In: Proc. CP, 17–35 (2005)

2. Crawford, J., Ginsberg, M., Luks, E., Roy, A.: Symmetry-breaking predicates for search problems. In: Proc. KR, 148–159 (1996)
3. Flener, P., Frisch, A.M., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Symmetry in matrix models. In: Proc. Symcon (2001)
4. Frisch, A.M., Harvey, W.: Constraints for breaking all row and column symmetries in a three-by-two-matrix. Proc. Symcon (2003)
5. Frisch, A.M., Jefferson, C., Martinez Hernandez, B., Miguel, I.: The rules of constraint modelling. Proc. IJCAI, 109–116 (2005)
6. Öhrman, H.: Breaking symmetries in matrix models. Masters Thesis, Dept. Information Technology, Uppsala University (2005)
7. Nuutila, E.: Efficient transitive closure computation in large digraphs. PhD Thesis, Acta Polytechnica Scandinavica, Mathematics and Computing in Engineering Series No. 74, Helsinki (1995)
8. Puget, J.-F.: On the satisfiability of symmetrical constraint satisfaction problems. In: Proc. ISMIS, 350–361 (1993)
9. Puget, J.-F.: Breaking symmetries in all different problems. In: Proc. IJCAI, 272–277 (2005)
10. Smith, B.M.: Sets of symmetry breaking constraints. In: Proc. Symcon (2005)