

Symmetry-Breaking Formulas for Groups with Bounded Orbit Projections

Amitabha Roy
Computer Science Dept,
Boston College.

September 22, 2007

The main contribution

Theorem:

- (i) If the symmetry group has orbits of fixed size there is a *complete* symmetry-breaking formula of size $O(n^6)$.

More generally

- (ii) If group projections in each orbit is bounded by n^d for fixed d (\mathcal{P}_d group) then there is a polynomial size SBF.

The main contribution

Theorem:

- (i) If the symmetry group has orbits of fixed size there is a *complete* symmetry-breaking formula of size $O(n^6)$.

More generally

- (ii) If group projections in each orbit is bounded by n^d for fixed d (\mathcal{P}_d group) then there is a polynomial size SBF.

Need to choose a special order of variables:

Variables in the same orbit appear together.

Previous Work

Theorem

- **Luks-R (2004)** If group orbits of size ≤ 2 , can find SBF of size $O(n^3)$ for every ordering ρ of Ω .

Previous Work

Theorem

- **Luks-R (2004)** If group orbits of size ≤ 2 , can find SBF of size $O(n^3)$ for every ordering ρ of Ω .
- **This work:**
If group has orbits of size k , there exists an ordering for which we can generate SBF of size $O(n^6)$.

Previous Work

Theorem

- **Luks-R (2004)** If group orbits of size ≤ 2 , can find SBF of size $O(n^3)$ for every ordering ρ of Ω .
- **This work:**
If group has orbits of size k , there exists an ordering for which we can generate SBF of size $O(n^6)$.
- **Luks-R (2004)** For abelian G , one can find ordering ρ and SBF of size $O(n^3 (\log n)^2)$.

Previous Work

Theorem

- **Luks-R (2004)** If group orbits of size ≤ 2 , can find SBF of size $O(n^3)$ for every ordering ρ of Ω .
- **This work:**
If group has orbits of size k , there exists an ordering for which we can generate SBF of size $O(n^6)$.
- **Luks-R (2004)** For abelian G , one can find ordering ρ and SBF of size $O(n^3 (\log n)^2)$.
- **Our work:**
Applies to non-abelian groups with small orbit projections.

Novelty of Current Approach

Previous Research

- Use computational group theory (nauty etc) to find symmetries.

Novelty of Current Approach

Previous Research

- Use computational group theory (nauty etc) to find symmetries.
- Use ad-hoc methods to generate SBFs from group. (prune naive lex-leader formula)

Novelty of Current Approach

Previous Research

- Use computational group theory (nauty etc) to find symmetries.
- Use ad-hoc methods to generate SBFs from group. (prune naive lex-leader formula)

Current Research:

- We use CGT to **generate** SBF!

Basic idea:

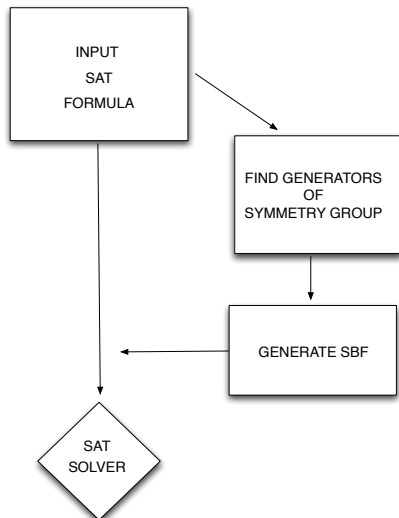
- Encode permutations and symmetries using Boolean variables and formulas.

Basic idea:

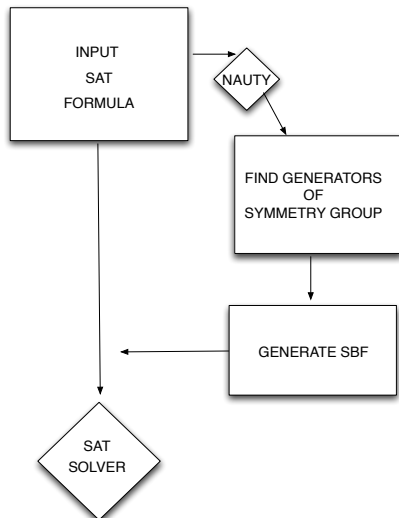
- Encode permutations and symmetries using Boolean variables and formulas.
- Algorithms in CGT are now Boolean formulas!

Algorithm we use: Sims's algorithm for testing membership in permutation groups.

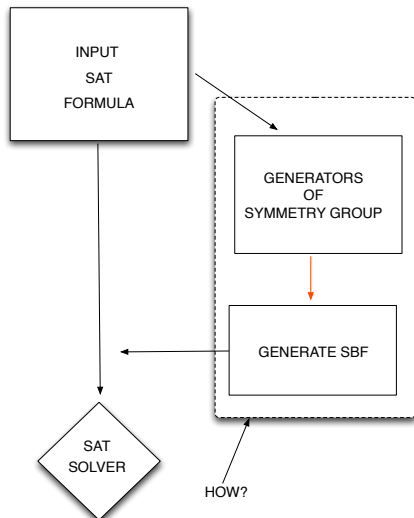
Review of the CGLR algorithm



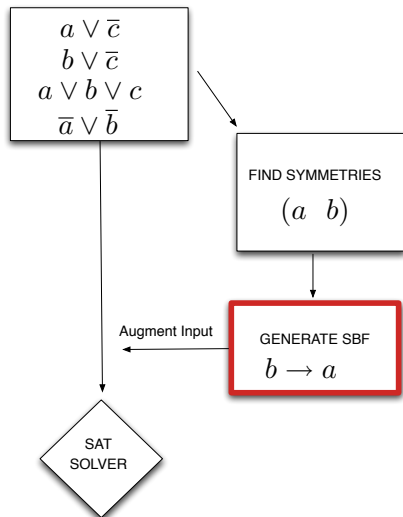
Review of the CGLR algorithm



Generating SBF



Generating SBF (Example)



Technical Details

The Algebraic Setup

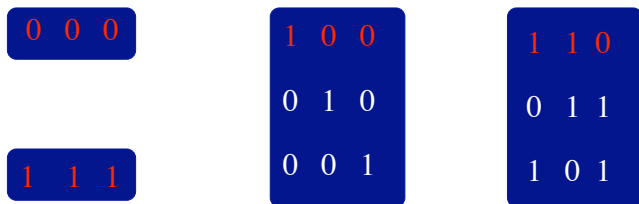
Input: $G = \langle X \rangle$ acting on $\Omega = \{1, 2, \dots, n\}$.

G also acts on n -bit strings by permuting coordinates:

$X \xrightarrow{g} X^g$ where $X^g(i) = X(i^g)$.

Set of all 2^n strings breaks into orbits.

$$G = \{(1\ 2\ 3), (1\ 3\ 2), ()\}$$



String Orbits on $\{1,2,3\}$ under G .



Group action on strings – example

$$G = \{(1\ 2\ 3), (1\ 3\ 2), ()\}$$

0 0 0

1 0 0

1 1 0

0 1 0

0 1 1

1 1 1

0 0 1

1 0 1

String Orbits on $\{1,2,3\}$ under G .

Need boolean formula over 3 variables, $X(1), X(2), X(3)$
satisfied by the red strings.

Example:

$$\begin{aligned} & (X(2) \rightarrow X(1)) \wedge (X(1) = X(2) \rightarrow (X(3) \rightarrow X(2))) \\ & (X(3) \rightarrow X(1)) \wedge (X(1) = X(3) \rightarrow (X(2) \rightarrow X(1))) \end{aligned}$$

Lex Leader Formula

Boolean formula $\Lambda(G, \rho)$ over variables $X(1), X(2) \dots X(n)$

May use additional variables.

Satisfying assignments of $\Lambda(G) =$ lexical leaders in each string orbit of G .

Need to construct $\Lambda(G, \rho)$ in poly time.

Additional Assumptions

Orbits of G are bounded.

Example

$$G = \langle (1, 2), (1, 2, 3, 4), (5, 6), (5, 6, 7) \rangle$$

Orbits are $\{1, 2, 3, 4\}, \{5, 6, 7\}$.

Assumption: Points in the same orbit are contiguous.

Strong Generating Set (SGS)

G_i = subgroup of G that fixes first i points.

Subgroup chain: $G = G_0 \geq G_1 \geq \cdots \geq G_{n-1} = 1$.

S = set of permutations

S is an SGS for G if $\langle S \cap G_i \rangle = G_i$.

Example:

$S = \{(1, 2, 3, 4), (2, 3, 4), (3, 4)\}$ SGS for S_4 since

$$\langle S \cap G_1 \rangle = \langle (2, 3, 4), (3, 4) \rangle = S_3$$

$$\langle S \cap G_2 \rangle = \langle (3, 4) \rangle = S_2$$

Sifting through SGS

R_i = complete set of right coset representatives of G_{i+1} in G_i .

Each coset representative \leftrightarrow point in the orbit of $i + 1$ in G_i .

Factorization process is called sifting.

If σ sifts through coset representatives, then $\sigma \in G$.

Possible to encode all this in the form of a Boolean formula.

Sifting in \mathcal{P}_d groups

$G = \langle R \rangle, R \leftarrow \text{SGS}$.

Given string X : define chain

$$\begin{aligned} G &\geq G_1^X \geq G_2^X \cdots \geq G_m^X \geq (G_m^X \cap G_1) \\ &\geq (G_m^X \cap G_2) \cdots \geq (G_m^X \cap G_{n-1} = 1) \end{aligned} \tag{1}$$

where $G_i^X = \{g \in G \mid g \text{ fixes } X_{\Delta_1}, X_{\Delta_2}, \dots, X_{\Delta_i}\}$.

Can represent SGS as a Boolean formula.

Basic Building Blocks

Permutation π represented by Boolean variables $\pi_{i,j}, 1 \leq i, j \leq n$.
Specify:

$$\bigvee_{j=1}^n \pi_{i,j} \equiv i \text{ is mapped to some point}$$

$$\bigwedge_{j \neq k} (\pi_{i,j} \rightarrow \neg \pi_{i,k}) \equiv i \text{ is mapped to unique point}$$

$O(n^2)$ formula. Too big!

Can be optimized using extra variables.

Basic Building Blocks (ctd)

Product(x, y, z) $\equiv x \cdot y = z$.

Member(L, π) $\equiv \pi \in L$.

Sift(π, S) $\equiv \pi$ sifts through S .

Basic Building Blocks (ctd)

Product(x, y, z) $\equiv x \cdot y = z$.

Member(L, π) $\equiv \pi \in L$.

Sift(π, S) $\equiv \pi$ sifts through S .

Lots of details! (read paper)

The main message

- Can encode CGT algorithms as SAT.
- Polytime algorithms = polynomial size formulas
- Unwieldy in practice

Effect of Order on Lex Leader Formula

Lex leader formula sensitive to order of permutation domain.

Open Question:

Are there groups G such that

- $\Lambda_{\text{nat}}(G)$ is of exponential size in one order ?
- $\Lambda_{\text{nat}}(G)$ is poly-size in some other order ?

Effect of Order (ctd)

Theorem (Babai-Luks)

Unless $P = NP$, there is no polynomial time algorithm that computes a lex leader formula $\Lambda(G)$ for an arbitrary group $G \leq \text{Sym}(\Omega)$.

Theorem indicates that there might be groups G where one can exhibit order sensitivity of $\Lambda(G)$.

Complexity of finding lex leaders for these groups depends on order of Ω (Babai, Luks):

- problem is **NP**-hard for some orders (decision version co-NP complete).
- problem is in **P** for some orders.

Future Work

- Make this practical.
- Theoretical extensions to other classes of groups (nilpotent, solvable)
- Effect of order.