Certified Static and Dynamic Symmetry Breaking

Bart Bogaerts
(Thanks to co-conspirators Jo Devriendt, Ward Gauderis,
Stephan Gocht, Ciaran McCreesh, Jakob Nordström)
Vrije Universiteit Brussel

Satisfiability: Theory, Practice, and Beyond
Simons Insitute; 20/04/2023

ARTIFICIAL
INTELLIGENCE
RESEARCH GROUP

INTRODUCTION

- I will assume familiarity with notions such as *literals*, *formulas*, SAT Solving, *CDCL*, ...
- I also assume everyone is convinced of the benefits of proof logging

INTRODUCTION

- I will assume familiarity with notions such as *literals*, *formulas*, SAT Solving, *CDCL*, ...
- I also assume everyone is convinced of the benefits of proof logging
- I will focus on symmetry handling:
  A permutation σ of literals is a (syntactic) symmetry of a formula F if:
  - σ respects negation: $\overline{\sigma(x)} = \sigma(\overline{x})$
  - $F\restriction_\sigma = F$
    ($F\restriction_\sigma$ is replacing each x by $\sigma(x)$ in F)

INTRODUCTION

- I will assume familiarity with notions such as *literals*, *formulas*, SAT Solving, *CDCL*, ...
- I also assume everyone is convinced of the benefits of proof logging
- I will focus on symmetry handling:
  A permutation σ of literals is a (syntactic) symmetry of a formula F if:
  - σ respects negation: $\overline{\sigma(x)} = \sigma(\overline{x})$
  - $F\restriction_\sigma = F$
    ($F\restriction_\sigma$ is replacing each x by $\sigma(x)$ in F)

Example

Consider the formula F:

$$\begin{array}{ll} a \lor \overline{b} \lor x \lor y & b \lor c \lor x \lor y \\ \overline{c} \lor \overline{d} \lor x \lor y & d \lor \overline{a} \lor x \lor y \end{array}$$

The permutation

$$(ab\overline{c}d)(xy)(\overline{a}\overline{b}c\overline{d})(\overline{x}\overline{y})$$

INTRODUCTION

- I will assume familiarity with notions such as *literals*, *formulas*, SAT Solving, *CDCL*, ...
- I also assume everyone is convinced of the benefits of proof logging
- I will focus on symmetry handling:
  A permutation σ of literals is a (syntactic) symmetry of a formula F if:
  - σ respects negation: $\overline{\sigma(x)} = \sigma(\overline{x})$
  - $F\restriction_\sigma = F$
    ($F\restriction_\sigma$ is replacing each x by $\sigma(x)$ in F)

Example

Consider the formula F:

$$\begin{array}{ll} a \lor \overline{b} \lor x \lor y & b \lor c \lor x \lor y \\ \overline{c} \lor \overline{d} \lor x \lor y & d \lor \overline{a} \lor x \lor y \end{array}$$

The permutation

$$(ab\overline{c}d)(xy)(\overline{a}\overline{b}c\overline{d})(\overline{x}\overline{y})$$

is a symmetry of F since $F\restriction_\sigma$ is

$$\begin{array}{ll} b \lor c \lor y \lor x & \overline{c} \lor \overline{d} \lor y \lor x \\ d \lor \overline{a} \lor y \lor x & a \lor \overline{b} \lor y \lor x \end{array}$$

INTRODUCTION

- I will assume familiarity with notions such as *literals*, *formulas*, SAT Solving, *CDCL*, ...
- I also assume everyone is convinced of the benefits of proof logging
- I will focus on symmetry handling:
  A permutation σ of literals is a (syntactic) symmetry of a formula F if:
  - σ respects negation: $\overline{\sigma(x)} = \sigma(\overline{x})$
  - $F\upharpoonright_\sigma = F$
    ($F\upharpoonright_\sigma$ is replacing each x by $\sigma(x)$ in F)
- Symmetric problems are often problematic for vanilla CDCL solvers (insert obligatory reference to PH principle here)

Example

Consider the formula F:

$$a \vee \overline{b} \vee x \vee y \qquad b \vee c \vee x \vee y$$

$$\overline{c} \vee \overline{d} \vee x \vee y \qquad d \vee \overline{a} \vee x \vee y$$

The permutation

$$(ab\overline{c}d)(xy)(\overline{a}\overline{b}c\overline{d})(\overline{x}\overline{y})$$

is a symmetry of F since $F\upharpoonright_\sigma$ is

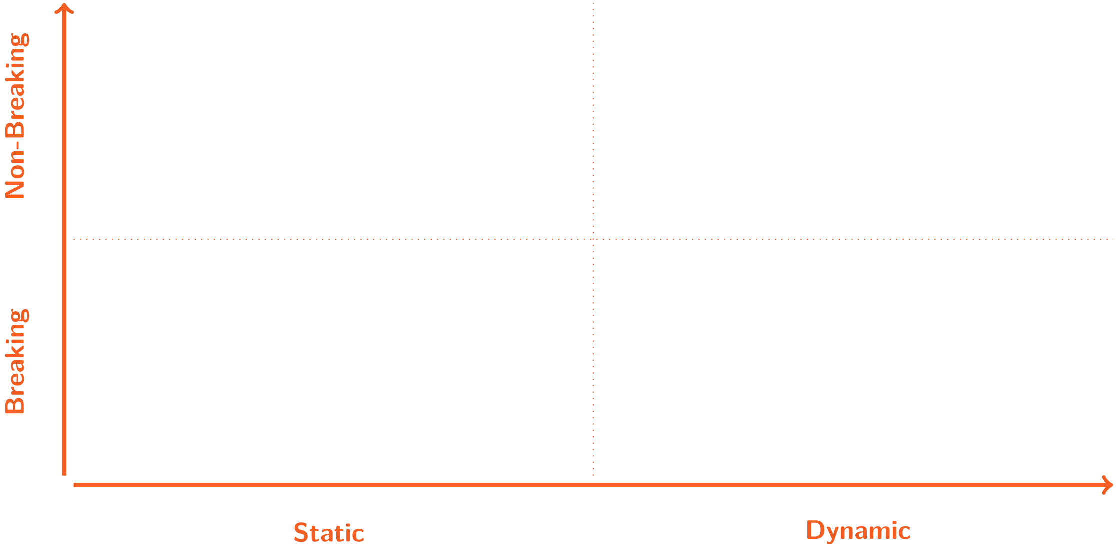
$$b \vee c \vee y \vee x \qquad \overline{c} \vee \overline{d} \vee y \vee x$$

$$d \vee \overline{a} \vee y \vee x \qquad a \vee \overline{b} \vee y \vee x$$

OUTLINE OF THIS TALK

1. Introduction
2. Handling Symmetries in SAT (Overview)
3. Symmetry Breaking with VeriPB
    1. The VeriPB proof System
    2. VeriPB-certified symmetry breaking
4. Conclusion

SYMMETRY HANDLING TECHNIQUES



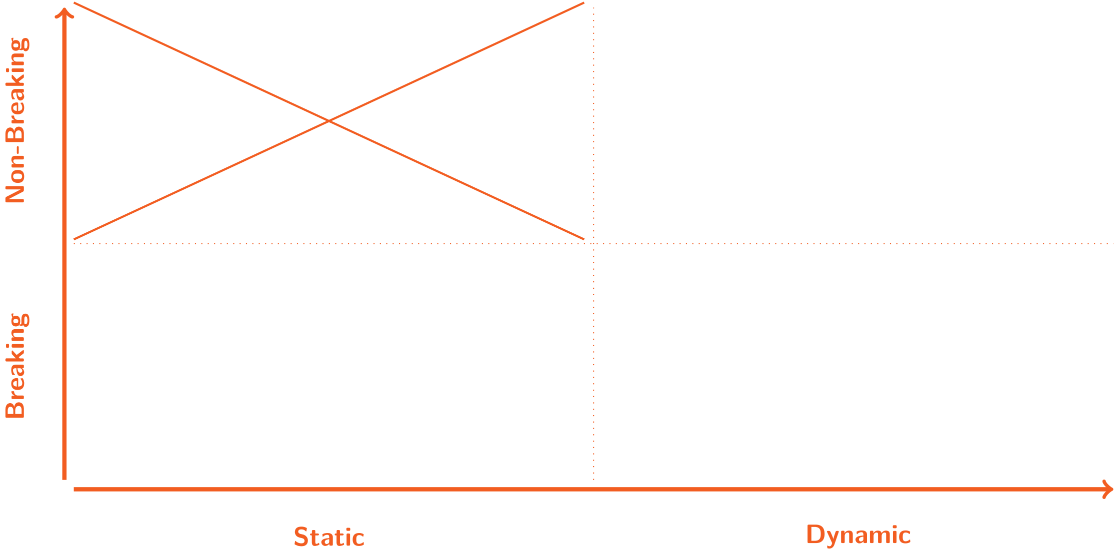
Non-Breaking

Breaking

Static

Dynamic

SYMMETRY HANDLING TECHNIQUES



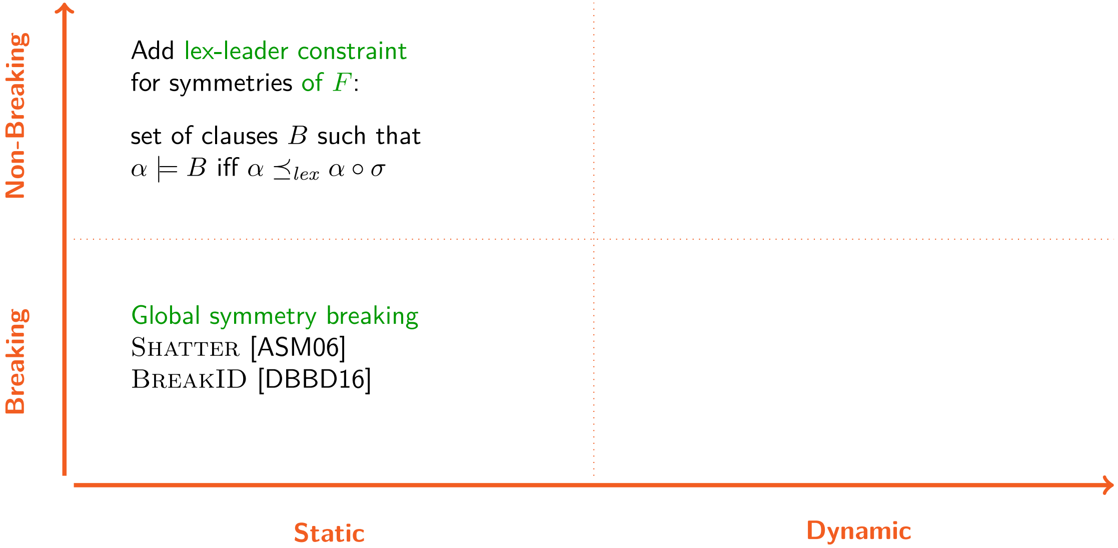
Non-Breaking

Breaking

Static

Dynamic

SYMMETRY HANDLING TECHNIQUES

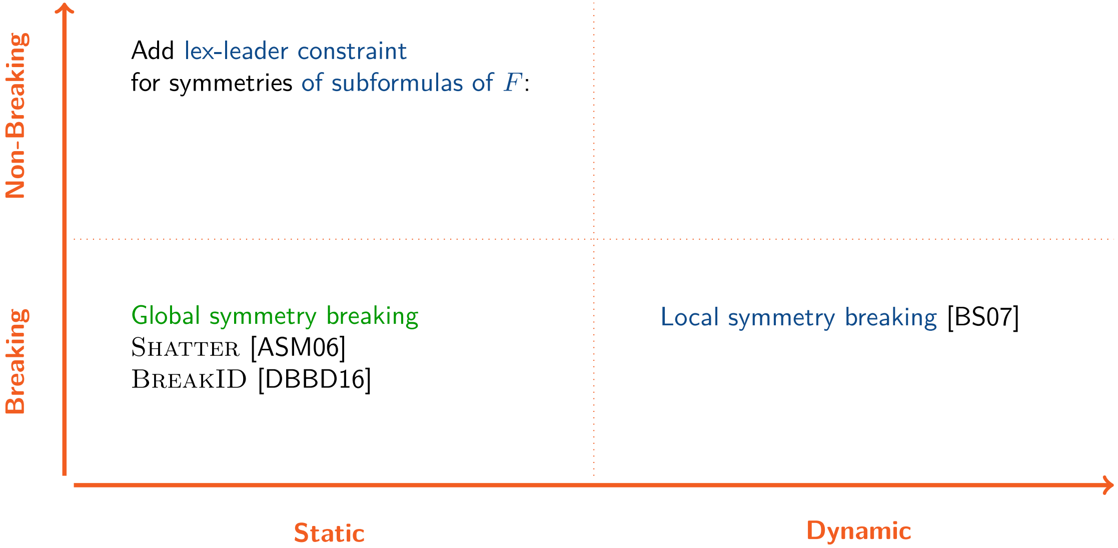


Add lex-leader constraint
for symmetries of F:

set of clauses B such that
$\alpha \models B$ iff $\alpha \preceq_{lex} \alpha \circ \sigma$

Global symmetry breaking
Shatter [ASM06]
BreakID [DBBD16]

SYMMETRY HANDLING TECHNIQUES

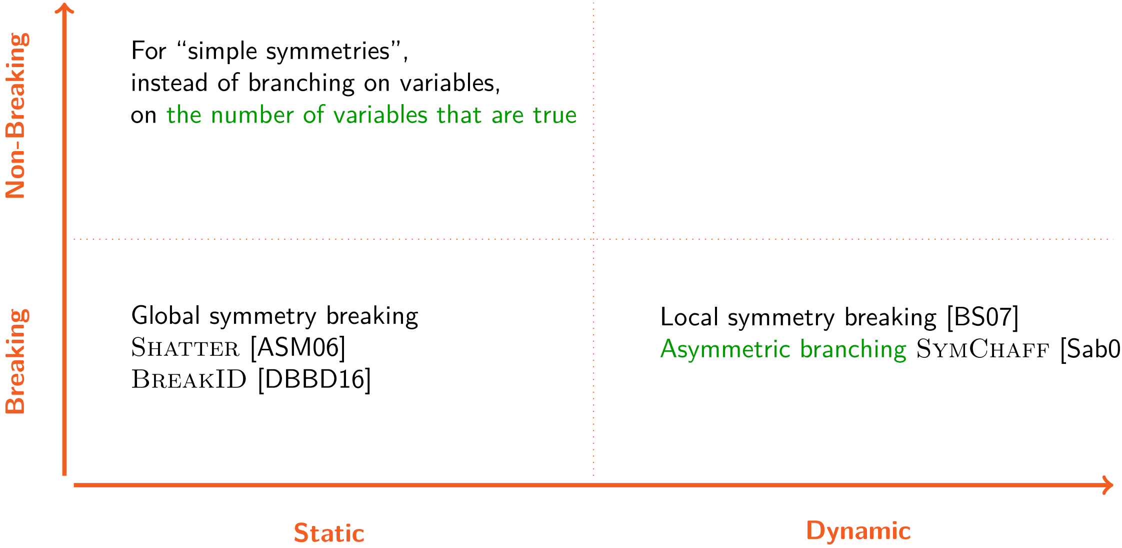


Add lex-leader constraint
for symmetries of subformulas of F:

Global symmetry breaking
SHATTER [ASM06]
BREAKID [DBBD16]

Local symmetry breaking [BS07]

SYMMETRY HANDLING TECHNIQUES

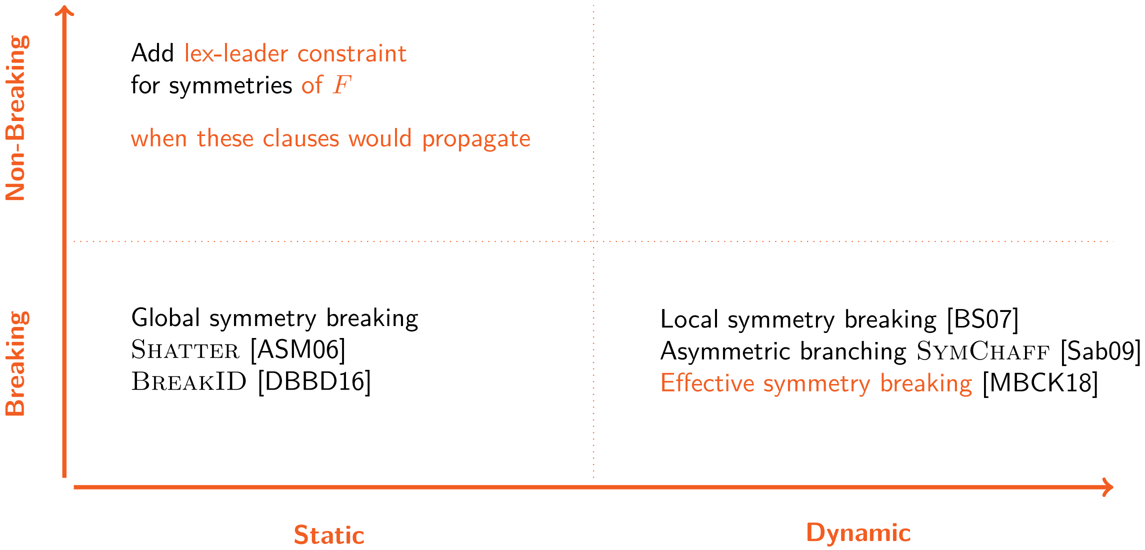


For "simple symmetries",
instead of branching on variables,
on the number of variables that are true

Global symmetry breaking
Shatter [ASM06]
BreakID [DBBD16]

Local symmetry breaking [BS07]
Asymmetric branching SymChaff [Sab09]

SYMMETRY HANDLING TECHNIQUES



Non-Breaking

Add lex-leader constraint
for symmetries of F

when these clauses would propagate

Breaking

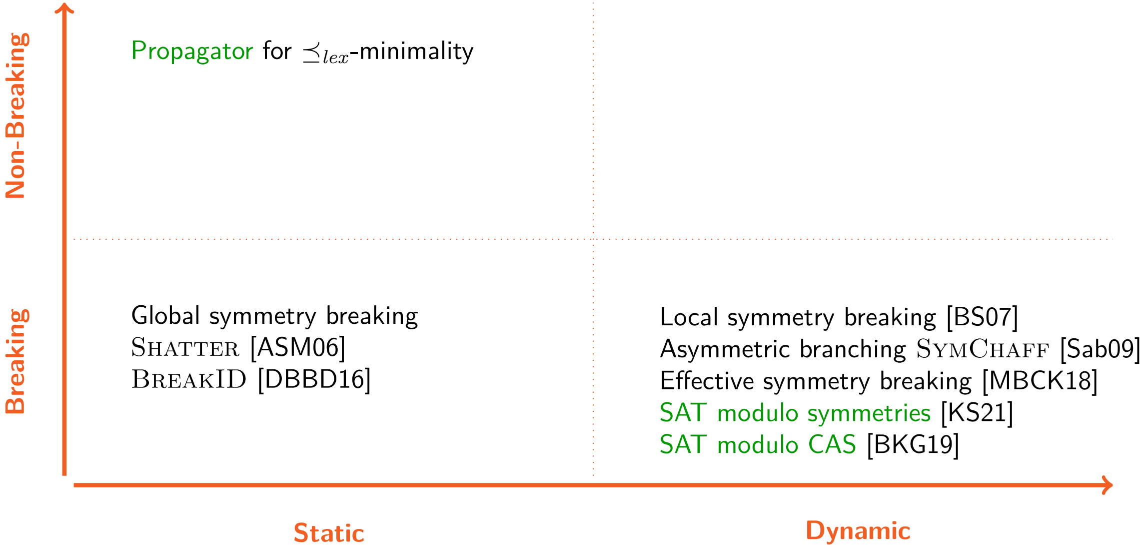
Global symmetry breaking
Shatter [ASM06]
BreakID [DBBD16]

Local symmetry breaking [BS07]
Asymmetric branching SymChaff [Sab09]
Effective symmetry breaking [MBCK18]

Static

Dynamic

SYMMETRY HANDLING TECHNIQUES



Propagator for $\preceq_{lex}$-minimality

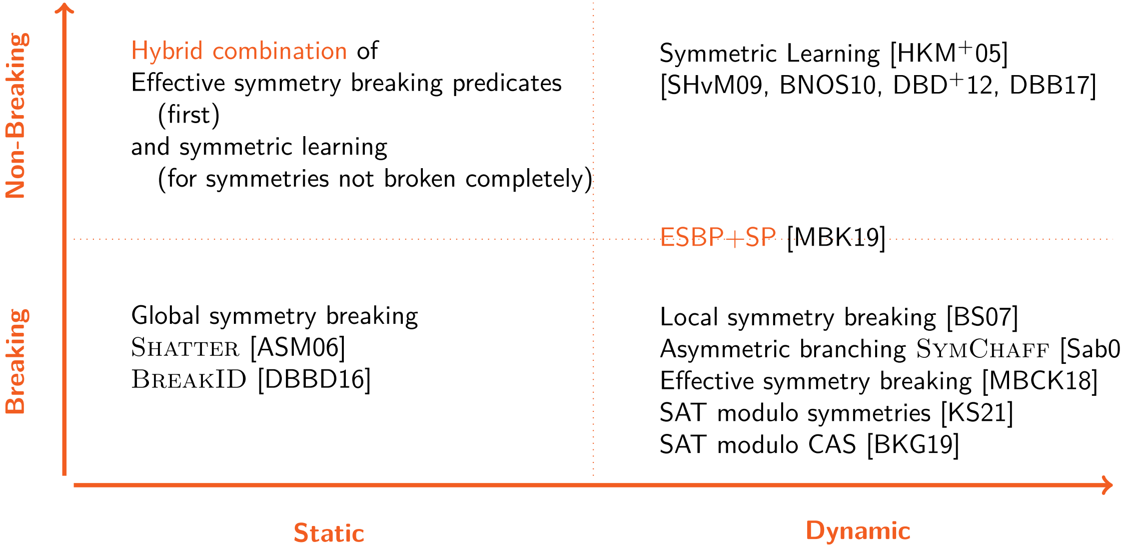
Global symmetry breaking
Shatter [ASM06]
BreakID [DBBD16]

Local symmetry breaking [BS07]
Asymmetric branching SymChaff [Sab09]
Effective symmetry breaking [MBCK18]
SAT modulo symmetries [KS21]
SAT modulo CAS [BKG19]

SYMMETRY HANDLING TECHNIQUES

	Static	Dynamic
Non-Breaking	When SAT solver learns c, also learn $c{\restriction}_{\sigma}$ (if this seems “interesting”)	Symmetric Learning [HKM^{+}05] [SHvM09, BNOS10, DBD^{+}12, DBB17]
Breaking	Global symmetry breaking Shatter [ASM06] BreakID [DBBD16]	Local symmetry breaking [BS07] Asymmetric branching SymChaff [Sab09] Effective symmetry breaking [MBCK18] SAT modulo symmetries [KS21] SAT modulo CAS [BKG19]

SYMMETRY HANDLING TECHNIQUES



Non-Breaking

Hybrid combination of
Effective symmetry breaking predicates
(first)
and symmetric learning
(for symmetries not broken completely)

Symmetric Learning [HKM$^+$05]
[SHvM09, BNOS10, DBD$^+$12, DBB17]

ESBP+SP [MBK19]

Breaking

Global symmetry breaking
Shatter [ASM06]
BreakID [DBBD16]

Local symmetry breaking [BS07]
Asymmetric branching SymChaff [Sab09]
Effective symmetry breaking [MBCK18]
SAT modulo symmetries [KS21]
SAT modulo CAS [BKG19]

Static

Dynamic

CERTIFIED SYMMETRY HANDLING

Static Symmetry breaking

- DRAT proof logging for limited cases only [HHW15]

CERTIFIED SYMMETRY HANDLING

Static Symmetry breaking

- DRAT proof logging for limited cases only [HHW15]
- VeriPB proof logging for general case [BGMN22]

CERTIFIED SYMMETRY HANDLING

Static Symmetry breaking

- DRAT proof logging for limited cases only [HHW15]
- VERIPB proof logging for general case [BGMN22]
- Also appears to be applicable to dynamic symmetry *breaking*

CERTIFIED SYMMETRY HANDLING

Static Symmetry breaking

- DRAT proof logging for limited cases only [HHW15]
- VeriPB proof logging for general case [BGMN22]
- Also appears to be applicable to dynamic symmetry *breaking* preliminary results
  - working version of ESBP [MBCK18]
  - revealed one or two bugs in implementation

CERTIFIED SYMMETRY HANDLING

Static Symmetry breaking

- DRAT proof logging for limited cases only [HHW15]
- VeriPB proof logging for general case [BGMN22]
- Also appears to be applicable to dynamic symmetry *breaking*
  preliminary results
  - working version of ESBP [MBCK18]
  - revealed one or two bugs in implementation

Symmetric learning

- Recently proposed proof logging [TD20]
  1. Special-purpose, specific approach
  2. Requires adding explicit concept of symmetries
  3. Not compatible with preprocessing techniques

  Better to keep proof system super-simple(?)

THE VERIPB PROOF SYSTEM

A proof system for **pseudo-Boolean optimization problems**

- Reasons with general pseudo-Boolean constraints
- Builds on cutting planes
- Extends this with strengthening rules (natural generalizations of RAT/PR)

THE VERIPB PROOF SYSTEM

A proof system for **pseudo-Boolean optimization problems**

- Reasons with general pseudo-Boolean constraints
- Builds on cutting planes
- Extends this with strengthening rules (natural generalizations of RAT/PR)

Details about the proof checker, see Stephan Gocht's PhD thesis [Goc22]

PSEUDO-BOOLEAN CONSTRAINTS

Pseudo-Boolean constraints are 0-1 integer linear constraints

$$\sum_i a_i \ell_i \geq A$$

- $a_i, A \in \mathbb{Z}$
- literals ℓ_i: x_i or $\overline{x}_i$ (where $x_i + \overline{x}_i = 1$)
- as before, variables x_i take values $0 = \mathit{false}$ or $1 = \mathit{true}$

PSEUDO-BOOLEAN REASONING: CUTTING PLANES [CCT87]

Literal axioms $$\frac{}{\ell_i \geq 0}$$

Linear combination $$\frac{\sum_i a_i \ell_i \geq A \qquad \sum_i b_i \ell_i \geq B}{\sum_i (c_A a_i + c_B b_i)\ell_i \geq c_A A + c_B B} \quad [c_A, c_B \in \mathbb{N}]$$

Division $$\frac{\sum_i c a_i \ell_i \geq A}{\sum_i a_i \ell_i \geq \lceil A/c \rceil} \quad [c \in \mathbb{N}^+]$$

REDUNDANCE-BASED STRENGTHENING

- C is redundant with respect to F if F and $F \wedge C$ are equisatisfiable
- Adding redundant constraints should be OK
- Notions such as RAT [JHB12] and propagation redundancy [HKB17]

REDUNDANCE-BASED STRENGTHENING

- C is redundant with respect to F if F and $F \wedge C$ are equisatisfiable
- Adding redundant constraints should be OK
- Notions such as RAT [JHB12] and propagation redundancy [HKB17]

Redundance-based strengthening [BT19, GN21]

C is redundant with respect to F if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$F \wedge \neg C \models (F \wedge C)\restriction_{\omega}$$

REDUNDANCE-BASED STRENGTHENING

Fact

$\alpha \models \phi\restriction_\omega$ *iff* $\alpha \circ \omega \models \phi$

- C is redundant with respect to F if F and $F \wedge C$ are equisatisfiable
- Adding redundant constraints should be OK
- Notions such as RAT [JHB12] and propagation redundancy [HKB17]

Redundance-based strengthening [BT19, GN21]

C is redundant with respect to F if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$F \wedge \neg C \models (F \wedge C)\restriction_\omega$$

REDUNDANCE-BASED STRENGTHENING

Fact

$\alpha \models \phi\restriction_\omega$ *iff* $\alpha \circ \omega \models \phi$

- C is redundant with respect to F if F and $F \wedge C$ are equisatisfiable
- Adding redundant constraints should be OK
- Notions such as RAT [JHB12] and propagation redundancy [HKB17]

Redundance-based strengthening [BT19, GN21]

C is redundant with respect to F if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$F \wedge \neg C \models (F \wedge C)\restriction_\omega$$

- Proof sketch for interesting direction: If α satisfies F but falsifies C, then $\alpha \circ \omega$ satisfies $F \wedge C$
- Implication should be efficiently verifiable (which is the case, e.g., if all constraints in $(F \wedge C)\restriction_\omega$ are RUP)

OPTIMIZATION PROBLEMS

Deal with symmetries by switching focus to optimization

OPTIMIZATION PROBLEMS

Deal with symmetries by switching focus to optimization

Pseudo-Boolean optimization

Minimize $f = \sum_i w_i \ell_i$ (for $w_i \in \mathbb{N}$) subject to constraints in F

OPTIMIZATION PROBLEMS

Deal with symmetries by switching focus to optimization

Pseudo-Boolean optimization

Minimize $f = \sum_i w_i \ell_i$ (for $w_i \in \mathbb{N}$) subject to constraints in F

Proof of optimality:

- F satisfied by α
- $F \wedge \left(\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)\right)$ is infeasible

OPTIMIZATION PROBLEMS

Deal with symmetries by switching focus to optimization

Pseudo-Boolean optimization

Minimize $f = \sum_i w_i \ell_i$ (for $w_i \in \mathbb{N}$) subject to constraints in F

Proof of optimality:

- F satisfied by α
- $F \wedge \left(\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)\right)$ is infeasible

Note that $\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)$ means $\sum_i w_i \ell_i \leq -1 + \sum_i w_i \cdot \alpha(\ell_i)$

OPTIMIZATION PROBLEMS

Deal with symmetries by switching focus to optimization

Pseudo-Boolean optimization

Minimize $f = \sum_i w_i \ell_i$ (for $w_i \in \mathbb{N}$) subject to constraints in F

Proof of optimality:

- F satisfied by α
- $F \wedge \left(\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)\right)$ is infeasible

Note that $\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)$ means $\sum_i w_i \ell_i \leq -1 + \sum_i w_i \cdot \alpha(\ell_i)$

Spoiler alert:
For decision problem, nothing stops us from inventing objective function
(like lexicographic order $\sum_{i=1}^{n} 2^i \cdot x_i$)

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?
Rules must preserve (at least one) optimal solution

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?
Rules must preserve (at least one) optimal solution

1. Standard cutting planes rules OK — derive constraints that must hold for any satisfying assignment

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?
Rules must preserve (at least one) optimal solution

1. Standard cutting planes rules OK — derive constraints that must hold for any satisfying assignment
2. Once solution α has been found, allow constraint $\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)$ to force search for better solutions

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?
Rules must preserve (at least one) optimal solution

1. Standard cutting planes rules OK — derive constraints that must hold for any satisfying assignment
2. Once solution α has been found, allow constraint $\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)$ to force search for better solutions
3. Redundance rule must not destroy good solutions

PROOF LOGGING FOR OPTIMIZATION PROBLEMS

How does proof system change?
Rules must preserve (at least one) optimal solution

1. Standard cutting planes rules OK — derive constraints that must hold for any satisfying assignment
2. Once solution α has been found, allow constraint $\sum_i w_i \ell_i < \sum_i w_i \cdot \alpha(\ell_i)$ to force search for better solutions
3. Redundance rule must not destroy good solutions

Redundance-based strengthening, optimization version [BGMN22]

Add constraint C to formula F if exists witness substitution ω such that

$$F \wedge \neg C \models (F \wedge C)\restriction_\omega \; \wedge \; f\restriction_\omega \leq f$$

REDUNDANCE AND DOMINANCE RULES

Redundance-based strengthening, optimization version [BGMN22]

Add constraint C to formula F if exists witness substitution ω such that

$$F \wedge \neg C \models (F \wedge C)\restriction_\omega \ \wedge \ f\restriction_\omega \leq f$$

REDUNDANCE AND DOMINANCE RULES

Redundance-based strengthening, optimization version [BGMN22]

Add constraint C to formula F if exists witness substitution ω such that

$$F \wedge \neg C \models (F \wedge C)\restriction_\omega \;\wedge\; f\restriction_\omega \leq f$$

Can be more aggressive if witness ω strictly improves solution

REDUNDANCE AND DOMINANCE RULES

Redundance-based strengthening, optimization version [BGMN22]

Add constraint C to formula F if exists witness substitution ω such that

$$F \land \neg C \models (F \land C)\restriction_\omega \land f\restriction_\omega \leq f$$

Can be more aggressive if witness ω strictly improves solution

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \land \neg D \models F\restriction_\omega \land f\restriction_\omega < f$$

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \wedge \; f\restriction_\omega < f$$

Why is this sound?

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \wedge f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \land \neg D \models F{\upharpoonright_\omega} \land f{\upharpoonright_\omega} < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \;\wedge\; f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \wedge f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies F and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \wedge f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies F and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies D, we're done

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \;\wedge\; f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies F and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies D, we're done
6. Otherwise $\big((\alpha \circ \omega) \circ \omega\big) \circ \omega$ satisfies F and $f\big(((\alpha \circ \omega) \circ \omega) \circ \omega\big) < f\big((\alpha \circ \omega) \circ \omega\big)$

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \;\wedge\; f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies F and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies D, we're done
6. Otherwise $\big((\alpha \circ \omega) \circ \omega\big) \circ \omega$ satisfies F and $f\big(((\alpha \circ \omega) \circ \omega) \circ \omega\big) < f\big((\alpha \circ \omega) \circ \omega\big)$
7. ...

SOUNDNESS OF DOMINANCE RULE

Dominance-based strengthening (simplified) [BGMN22]

Add constraint D to formula F if exists witness substitution ω such that

$$F \wedge \neg D \models F\restriction_\omega \ \wedge \ f\restriction_\omega < f$$

Why is this sound?

1. Suppose α satisfies F but falsifies D (i.e., satisfies $\neg D$)
2. Then $\alpha \circ \omega$ satisfies F and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies D, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies F and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies D, we're done
6. Otherwise $\big((\alpha \circ \omega) \circ \omega\big) \circ \omega$ satisfies F and $f\big(((\alpha \circ \omega) \circ \omega) \circ \omega\big) < f\big((\alpha \circ \omega) \circ \omega\big)$
7. . . .
8. Can't go on forever, so finally reach α' satisfying $F \wedge D$

STRENGTH OF DOMINANCE RULE

Dominance-based strengthening (stronger, still simplified) [BGMN22]

If $D_1, D_2, \ldots, D_{m-1}$ have been derived from F (maybe using dominance), then can derive also D_m if exists witness substitution ω such that

$$F \wedge \bigwedge_{i=1}^{m-1} D_i \wedge \neg D_m \models F{\upharpoonright_\omega} \wedge f{\upharpoonright_\omega} < f$$

STRENGTH OF DOMINANCE RULE

Dominance-based strengthening (stronger, still simplified) [BGMN22]

If $D_1, D_2, \ldots, D_{m-1}$ have been derived from F (maybe using dominance), then can derive also D_m if exists witness substitution ω such that

$$F \wedge \bigwedge_{i=1}^{m-1} D_i \wedge \neg D_m \models F\restriction_\omega \wedge f\restriction_\omega < f$$

Why is this sound?

- Same inductive proof as before, but nested

STRENGTH OF DOMINANCE RULE

Dominance-based strengthening (stronger, still simplified) [BGMN22]

If $D_1, D_2, \ldots, D_{m-1}$ have been derived from F (maybe using dominance), then can derive also D_m if exists witness substitution ω such that

$$F \wedge \bigwedge_{i=1}^{m-1} D_i \wedge \neg D_m \models F\!\upharpoonright_\omega \wedge f\!\upharpoonright_\omega < f$$

Why is this sound?

- Same inductive proof as before, but nested
- Or just pick α satisfying F and minimizing f and argue by contradiction

STRENGTH OF DOMINANCE RULE

Dominance-based strengthening (stronger, still simplified) [BGMN22]

If $D_1, D_2, \ldots, D_{m-1}$ have been derived from F (maybe using dominance), then can derive also D_m if exists witness substitution ω such that

$$F \wedge \bigwedge_{i=1}^{m-1} D_i \wedge \neg D_m \models F{\restriction_\omega} \wedge f{\restriction_\omega} < f$$

Why is this sound?

- Same inductive proof as before, but nested
- Or just pick α satisfying F and minimizing f and argue by contradiction

Further extensions:

- Define dominance rule w.r.t. order independent of objective function

STRENGTH OF DOMINANCE RULE

Dominance-based strengthening (stronger, still simplified) [BGMN22]

If $D_1, D_2, \ldots, D_{m-1}$ have been derived from F (maybe using dominance), then can derive also D_m if exists witness substitution ω such that

$$F \wedge \bigwedge_{i=1}^{m-1} D_i \wedge \neg D_m \models F\restriction_\omega \wedge f\restriction_\omega < f$$

Why is this sound?

- Same inductive proof as before, but nested
- Or just pick α satisfying F and minimizing f and argue by contradiction

Further extensions:

- Define dominance rule w.r.t. order independent of objective function
- Switch between different orders in same proof

STRATEGY FOR SAT SYMMETRY BREAKING

1. Pretend to solve optimisation problem minimizing $f \doteq \sum_{i=1}^{n} 2^{n-i} \cdot x_i$
   (searching lexicographically smallest assignment satisfying formula)

STRATEGY FOR SAT SYMMETRY BREAKING

1. Pretend to solve optimisation problem minimizing $f \doteq \sum_{i=1}^{n} 2^{n-i} \cdot x_i$ (searching lexicographically smallest assignment satisfying formula)
2. Derive pseudo-Boolean lex-leader constraint

$$
\begin{aligned}
C_\sigma &\doteq f \leq f\restriction_\sigma \\
&\doteq \sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0
\end{aligned}
$$

STRATEGY FOR SAT SYMMETRY BREAKING

1. Pretend to solve optimisation problem minimizing $f \doteq \sum_{i=1}^{n} 2^{n-i} \cdot x_i$ (searching lexicographically smallest assignment satisfying formula)
2. Derive pseudo-Boolean lex-leader constraint

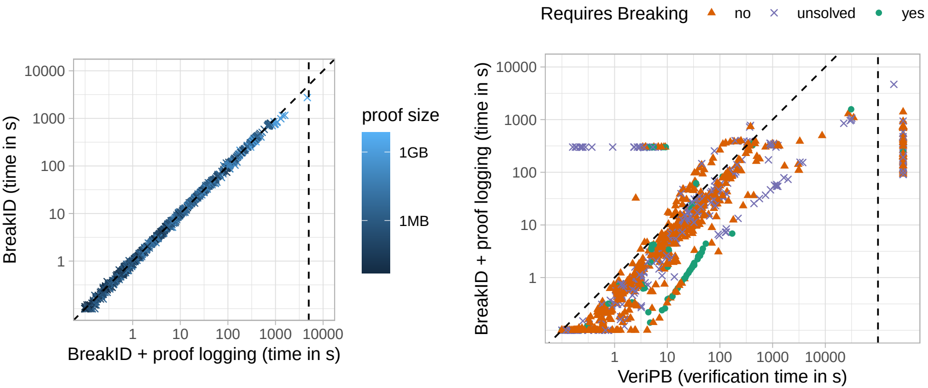
$$\begin{aligned} C_\sigma &\doteq f \leq f\restriction_\sigma \\ &\doteq \sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0 \end{aligned}$$

3. Derive CNF encoding of lex-leader constraints from PB constraint (in same spirit as [GMNO22])

$$\begin{aligned} &y_0 && \overline{y}_j \vee \overline{\sigma(x_j)} \vee x_j \\ &\overline{y}_{j-1} \vee \overline{x}_j \vee \sigma(x_j) && y_j \vee \overline{y}_{j-1} \vee \overline{x}_j \\ &\overline{y}_j \vee y_{j-1} && y_j \vee \overline{y}_{j-1} \vee \sigma(x_j) \end{aligned}$$

EXPERIMENTAL EVALUATION

- Evaluated on SAT competition benchmarks
- BREAKID [DBBD16, Bre] used to find and break symmetries



- proof logging overhead negligible
- verification at most 20 times slower than solving for 95% of instances

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (1/2)

Definition

Given a symmetry σ, the (pseudo-Boolean) breaking constraint of σ is

$$C_\sigma \doteq f \leq f\restriction_\sigma$$

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (1/2)

Definition

Given a symmetry σ, the (pseudo-Boolean) breaking constraint of σ is

$$C_\sigma \doteq f \leq f\upharpoonright_\sigma$$

Theorem

C_σ can be derived from F using dominance with witness σ

$$F \wedge \neg C_\sigma \models F\upharpoonright_\sigma \wedge f\upharpoonright_\sigma < f$$

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (2/2)

Breaking symmetries with the dominance rule

- Surprisingly simple

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (2/2)

Breaking symmetries with the dominance rule

- Surprisingly simple
- Generalizes well

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (2/2)

Breaking symmetries with the dominance rule

- Surprisingly simple
- Generalizes well
  - Works for arbitrary symmetries

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (2/2)

Breaking symmetries with the dominance rule

- Surprisingly simple
- Generalizes well
  - Works for arbitrary symmetries
  - Works for multiple symmetries (ignore previously derived constraints)

$$F \wedge C_{12} \wedge \neg C_{23} \models F\restriction_{\sigma_{(23)}} \wedge f\restriction_{\sigma_{(23)}} < f$$

BREAKING SYMMETRIES WITH THE DOMINANCE RULE (2/2)

Breaking symmetries with the dominance rule

- Surprisingly simple
- Generalizes well
    - Works for arbitrary symmetries
    - Works for multiple symmetries (ignore previously derived constraints)

$$F \land C_{12} \land \neg C_{23} \models F\restriction_{\sigma_{(23)}} \land f\restriction_{\sigma_{(23)}} < f$$

Why does it work?

- Witness need not satisfy all derived constraints
- Sufficient to just produce “better” assignment

CONCLUSION

- Variety of symmetry handling methods

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VeriPB

  Challenge: get this to work in (some extension of) DRAT

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VeriPB
  Challenge: get this to work in (some extension of) DRAT
- Makes use heavily of dominance rule
  Challenge: analyze this rule (can extended Frege simulate it?)

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VERIPB

  Challenge: get this to work in (some extension of) DRAT
- Makes use heavily of dominance rule

  Challenge: analyze this rule (can extended Frege simulate it?)
- Claim that this generalizes to dynamic symmetry breaking methods

  Challenge: Verify this for other dynamic symmetry breaking methods

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VERIPB

  Challenge: get this to work in (some extension of) DRAT
- Makes use heavily of dominance rule

  Challenge: analyze this rule (can extended Frege simulate it?)
- Claim that this generalizes to dynamic symmetry breaking methods

  Challenge: Verify this for other dynamic symmetry breaking methods
- For symmetric learning, dedicated proof system has been developed

  Challenge: develop certification in a formalism that doesn't know about symmetries

  Proofs with lemmas?

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VeriPB
  Challenge: get this to work in (some extension of) DRAT
- Makes use heavily of dominance rule
  Challenge: analyze this rule (can extended Frege simulate it?)
- Claim that this generalizes to dynamic symmetry breaking methods
  Challenge: Verify this for other dynamic symmetry breaking methods
- For symmetric learning, dedicated proof system has been developed
  Challenge: develop certification in a formalism that doesn't know about symmetries
  Proofs with lemmas?

CONCLUSION

- Variety of symmetry handling methods
- For static (and dynamic) symmetry breaking, fully general symmetry breaking in VeriPB
  Challenge: get this to work in (some extension of) DRAT
- Makes use heavily of dominance rule
  Challenge: analyze this rule (can extended Frege simulate it?)
- Claim that this generalizes to dynamic symmetry breaking methods
  Challenge: Verify this for other dynamic symmetry breaking methods
- For symmetric learning, dedicated proof system has been developed
  Challenge: develop certification in a formalism that doesn't know about symmetries
  Proofs with lemmas?

Thank you for your attention!

SYMMETRY BREAKING: EXAMPLE

Example (Pigeonhole principle formula)

- Variables p_{ij} $(1 \leq i \leq 4, 1 \leq j \leq 3)$ true iff pigeon i in hole j
- Focus on pigeon symmetries — notation:
  - $\sigma_{(12)}$ swaps pigeons 1 and 2

SYMMETRY BREAKING: EXAMPLE

Example (Pigeonhole principle formula)

- Variables p_{ij} $(1 \leq i \leq 4, 1 \leq j \leq 3)$ true iff pigeon i in hole j
- Focus on pigeon symmetries — notation:
  - $\sigma_{(12)}$ swaps pigeons 1 and 2
    Formally: $\sigma_{(12)}(p_{1j}) = p_{2j}$ and $\sigma_{(12)}(p_{2j}) = p_{1j}$ for all j
  - $\sigma_{(1234)}$ shifts all pigeons

SYMMETRY BREAKING: EXAMPLE

Example (Pigeonhole principle formula)

- Variables p_{ij} $(1 \leq i \leq 4, 1 \leq j \leq 3)$ true iff pigeon i in hole j
- Focus on pigeon symmetries — notation:
    - $\sigma_{(12)}$ swaps pigeons 1 and 2
      Formally: $\sigma_{(12)}(p_{1j}) = p_{2j}$ and $\sigma_{(12)}(p_{2j}) = p_{1j}$ for all j
    - $\sigma_{(1234)}$ shifts all pigeons

Order: "Pigeon 1 preferred in the largest possible hole; next pigeon 2, ..."

$$f \doteq 2^{11} \cdot p_{13} + 2^{10} \cdot p_{12} + 2^{9} \cdot p_{11} + 2^{8} \cdot p_{23} + \cdots + 1 \cdot p_{41}$$

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F{\upharpoonright}_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F\restriction_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

$$C_{12} \doteq f \leq f\restriction_{\sigma_{(12)}}$$

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F\restriction_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

$$\begin{aligned} C_{12} &\doteq f \leq f\restriction_{\sigma_{(12)}} \\ &\doteq \sum_{i=1}^{n} 2^{n-i} \cdot \left(\sigma_{(12)}(x_i) - x_i\right) \geq 0 \end{aligned}$$

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F\restriction_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

$$\begin{aligned} C_{12} &\doteq f \leq f\restriction_{\sigma_{(12)}} \\ &\doteq \textstyle\sum_{i=1}^{n} 2^{n-i} \cdot \big(\sigma_{(12)}(x_i) - x_i\big) \geq 0 \\ &\doteq \big(2^{11}-2^{8}\big)(p_{23}-p_{13}) + \big(2^{10}-2^{7}\big)(p_{22}-p_{12}) + \big(2^{9}-2^{6}\big)(p_{21}-p_{11}) \geq 0 \end{aligned}$$

"Pigeon 1 in smaller hole than pigeon 2"

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F\restriction_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

$$\begin{aligned} C_{12} &\doteq f \leq f\restriction_{\sigma_{(12)}} \\ &\doteq \sum_{i=1}^{n} 2^{n-i} \cdot \left(\sigma_{(12)}(x_i) - x_i\right) \geq 0 \\ &\doteq \left(2^{11} - 2^{8}\right)(p_{23} - p_{13}) + \left(2^{10} - 2^{7}\right)(p_{22} - p_{12}) + \left(2^{9} - 2^{6}\right)(p_{21} - p_{11}) \geq 0 \end{aligned}$$

"Pigeon 1 in smaller hole than pigeon 2"

- Can be added with redundance rule (the symmetry is the witness):

$$F \wedge \neg C_{12} \models F\restriction_{\sigma_{(12)}} \wedge C_{12}\restriction_{\sigma_{(12)}} \wedge f\restriction_{\sigma_{(12)}} \leq f$$

$$F \wedge f > f\restriction_{\sigma_{(12)}} \models F\restriction_{\sigma_{(12)}} \wedge f \geq f\restriction_{\sigma_{(12)}} \wedge f\restriction_{\sigma_{(12)}} \leq f$$

BREAKING A SINGLE SIMPLE SYMMETRY (EXAMPLE)

- F is a formula expressing PHP constraints with $F\upharpoonright_{\sigma_{(12)}} = F$
- Want to add constraint C_{12} breaking $\sigma_{(12)}$ — should be satisfied by α iff α "at least as good" as $\sigma_{(12)}(\alpha)$

$$\begin{aligned} C_{12} &\doteq f \leq f\upharpoonright_{\sigma_{(12)}} \\ &\doteq \sum_{i=1}^{n} 2^{n-i} \cdot \big(\sigma_{(12)}(x_i) - x_i\big) \geq 0 \\ &\doteq \big(2^{11} - 2^{8}\big)(p_{23} - p_{13}) + \big(2^{10} - 2^{7}\big)(p_{22} - p_{12}) + \big(2^{9} - 2^{6}\big)(p_{21} - p_{11}) \geq 0 \end{aligned}$$

"Pigeon 1 in smaller hole than pigeon 2"

- Can be added with redundance rule (the symmetry is the witness):

$$F \wedge \neg C_{12} \models F\upharpoonright_{\sigma_{(12)}} \wedge C_{12}\upharpoonright_{\sigma_{(12)}} \ \wedge\ f\upharpoonright_{\sigma_{(12)}} \leq f$$

$$F \wedge f > f\upharpoonright_{\sigma_{(12)}} \models F\upharpoonright_{\sigma_{(12)}} \wedge f \geq f\upharpoonright_{\sigma_{(12)}} \ \wedge\ f\upharpoonright_{\sigma_{(12)}} \leq f$$

Similar to DRAT symmetry breaking [HHW15]

BREAKING MORE/OTHER SYMMETRIES

Problem

This idea does not generalize.

- Breaking two symmetries

- Breaking complex symmetries

BREAKING MORE/OTHER SYMMETRIES

Problem

This idea does not generalize.

- Breaking two symmetries

$$F \land C_{12} \land \neg C_{23} \not\models F\restriction_{\sigma_{(23)}} \land C_{12}\restriction_{\sigma_{(23)}} \land C_{23}\restriction_{\sigma_{(23)}} \land f\restriction_{\sigma_{(23)}} \leq f$$

Intuitively: applying $\sigma_{(23)}$ potentially falsifies C_{12}

- Breaking complex symmetries

BREAKING MORE/OTHER SYMMETRIES

Problem

This idea does not generalize.

- Breaking two symmetries

$$F \wedge C_{12} \wedge \neg C_{23} \not\models F\restriction_{\sigma_{(23)}} \wedge C_{12}\restriction_{\sigma_{(23)}} \wedge C_{23}\restriction_{\sigma_{(23)}} \wedge f\restriction_{\sigma_{(23)}} \leq f$$

  Intuitively: applying $\sigma_{(23)}$ potentially falsifies C_{12}
  We might have to apply $\sigma_{(12)}$ again

- Breaking complex symmetries

BREAKING MORE/OTHER SYMMETRIES

Problem

This idea does not generalize.

- Breaking two symmetries

$$F \wedge C_{12} \wedge \neg C_{23} \not\models F\restriction_{\sigma_{(23)}} \wedge C_{12}\restriction_{\sigma_{(23)}} \wedge C_{23}\restriction_{\sigma_{(23)}} \ \wedge \ f\restriction_{\sigma_{(23)}} \leq f$$

  Intuitively: applying $\sigma_{(23)}$ potentially falsifies C_{12}
  We might have to apply $\sigma_{(12)}$ again

- Breaking complex symmetries

$$F \wedge \neg C_{1234} \models F\restriction_{\sigma_{(1234)}} \wedge C_{1234}\restriction_{\sigma_{(1234)}} \ \wedge \ f\restriction_{\sigma_{(1234)}} \leq f$$

  Intuitively, C_{1234} holds if shifting all the pigeons results in a worse assignment.

BREAKING MORE/OTHER SYMMETRIES

Problem

This idea does not generalize.

- Breaking two symmetries

$$F \wedge C_{12} \wedge \neg C_{23} \not\models F\restriction_{\sigma_{(23)}} \wedge C_{12}\restriction_{\sigma_{(23)}} \wedge C_{23}\restriction_{\sigma_{(23)}} \wedge f\restriction_{\sigma_{(23)}} \leq f$$

Intuitively: applying $\sigma_{(23)}$ potentially falsifies C_{12}
We might have to apply $\sigma_{(12)}$ again

- Breaking complex symmetries

$$F \wedge \neg C_{1234} \models F\restriction_{\sigma_{(1234)}} \wedge C_{1234}\restriction_{\sigma_{(1234)}} \wedge f\restriction_{\sigma_{(1234)}} \leq f$$

Intuitively, C_{1234} holds if shifting all the pigeons results in a worse assignment.
If it is falsified, we can “restore” its truth by applying $\sigma_{(1234)}$ once, twice, or thrice.

STRATEGY FOR SAT SYMMETRY BREAKING

1. Pretend to solve optimisation problem minimizing $f \doteq \sum_{i=1}^{n} 2^{n-i} \cdot x_i$ (searching lexicographically smallest assignment satisfying formula)
2. Derive pseudo-Boolean lex-leader constraint

$$\begin{aligned} C_\sigma &\doteq f \leq f\restriction_\sigma \\ &\doteq \sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0 \end{aligned}$$

3. Derive CNF encoding of lex-leader constraints from PB constraint (in same spirit as [GMNO22])

$$\begin{array}{ll} y_0 & \overline{y}_j \vee \overline{\sigma(x_j)} \vee x_j \\ \overline{y}_{j-1} \vee \overline{x}_j \vee \sigma(x_j) & y_j \vee \overline{y}_{j-1} \vee \overline{x}_j \\ \overline{y}_j \vee y_{j-1} & y_j \vee \overline{y}_{j-1} \vee \sigma(x_j) \end{array}$$

SYMMETRY BREAKING IN CNF

- In SAT symmetry breakers, symmetry is broken in CNF

SYMMETRY BREAKING IN CNF

- In SAT symmetry breakers, symmetry is broken in CNF
- Still need to show how to derive CNF encoding

SYMMETRY BREAKING IN CNF

- In SAT symmetry breakers, symmetry is broken in CNF
- Still need to show how to derive CNF encoding
- We use the encoding of BreakID [DBBD16]:

$$
\begin{aligned}
&y_0\\
&\overline{y}_{j-1} \vee \overline{x}_j \vee \sigma(x_j)\\
&\overline{y}_j \vee y_{j-1}\\
&\overline{y}_j \vee \overline{\sigma(x_j)} \vee x_j\\
&y_j \vee \overline{y}_{j-1} \vee \overline{x}_j\\
&y_j \vee \overline{y}_{j-1} \vee \sigma(x_j)
\end{aligned}
$$

SYMMETRY BREAKING IN CNF

- In SAT symmetry breakers, symmetry is broken in CNF
- Still need to show how to derive CNF encoding
- We use the encoding of BreakID [DBBD16]:

$$y_0$$

$$\overline{y}_{j-1} \vee \overline{x}_j \vee \sigma(x_j)$$

$$\overline{y}_j \vee y_{j-1}$$

$$\overline{y}_j \vee \overline{\sigma(x_j)} \vee x_j$$

$$y_j \vee \overline{y}_{j-1} \vee \overline{x}_j$$

$$y_j \vee \overline{y}_{j-1} \vee \sigma(x_j)$$

Define y_j to be true if x_k equals $\sigma(x_k)$ for all $k \leq j$

$$y_k \Leftrightarrow y_{k-1} \wedge (x_k \Leftrightarrow \sigma(x_k))$$

(derivable with redundance rule)

SYMMETRY BREAKING IN CNF

- In SAT symmetry breakers, symmetry is broken in CNF
- Still need to show how to derive CNF encoding
- We use the encoding of BreakID [DBBD16]:

$$y_0$$

$$\overline{y}_{j-1} \vee \overline{x}_j \vee \sigma(x_j)$$

$$\overline{y}_j \vee y_{j-1}$$

$$\overline{y}_j \vee \overline{\sigma(x_j)} \vee x_j$$

$$y_j \vee \overline{y}_{j-1} \vee \overline{x}_j$$

$$y_j \vee \overline{y}_{j-1} \vee \sigma(x_j)$$

Define y_j to be true if x_k equals $\sigma(x_k)$ for all $k \leq j$

$$y_k \Leftrightarrow y_{k-1} \wedge (x_k \Leftrightarrow \sigma(x_k))$$

(derivable with redundance rule) If y_k is true, x_k is at most $\sigma(x_k)$
(derivable from the PB breaking constraint)

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

Pseudo-Boolean breaking constraint

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

Derivable by redundance with witness $\omega : y_0 \mapsto 1$

$$F \wedge D \wedge \{\overline{y}_0\} \models (F \wedge D)\restriction_\omega \wedge \{y_0\}\restriction_\omega$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

Derivable by redundance with witness $\omega : y_0 \mapsto 1$

$$F \wedge D \wedge \{\overline{y}_0\} \models (F \wedge D)\restriction_\omega \wedge \{y_0\}\restriction_\omega$$

$$F \wedge \{\overline{y}_0\} \models (F \wedge D)\restriction_\omega \wedge \{1\}$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

Derivable by RUP

$$F \wedge D \wedge \neg(\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1))$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

Derivable by RUP

$$F \wedge D \wedge \neg(\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1))$$

$$= F \wedge D \wedge \{y_0 \wedge x_1 \wedge \overline{\sigma(x_1)}\}$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

Derivable by RUP

$$F \wedge D \wedge \neg(\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1))$$
$$= F \wedge D \wedge \{y_0 \wedge x_1 \wedge \overline{\sigma(x_1)}\}$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

Derivable by RUP

$$F \wedge D \wedge \neg(\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1))$$
$$= F \wedge D \wedge \{y_0 \wedge x_1 \wedge \overline{\sigma(x_1)}\}$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$2^{n-1} \cdot (-1) + \sum_{i=2}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

Derivable by RUP

$$F \wedge D \wedge \neg(\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1))$$
$$= F \wedge D \wedge \{y_0 \wedge x_1 \wedge \overline{\sigma(x_1)}\}$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$2^{n-1} \cdot (-1) + \sum_{i=2}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

with

$$\sum_{i=2}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \leq 2^{n-1} - 1$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

$\overline{y}_1 \vee y_0$

Derivable by redundance with witness $\omega : y_1 \mapsto 0$

$$F \wedge D \wedge \neg(\overline{y}_1 \vee y_0) \models (F \wedge D)\restriction_\omega \wedge \{\overline{y}_1 \vee y_0\}\restriction_\omega$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

Derivable by redundance with witness $\omega : y_1 \mapsto 0$

$$F \wedge D \wedge \neg(\overline{y}_1 \vee y_0)$$

$$\models (F \wedge D)\restriction_\omega \wedge \{\overline{y}_1 \vee y_0\}\restriction_\omega$$

$$F \wedge D \wedge \neg(\overline{y}_1 \vee y_0)$$

$$\models (F \wedge D)\restriction_\omega \wedge \{1 \vee y_0\}$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

$\overline{y}_1 \vee y_0$

$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$

Derivable by redundance with witness $\omega : y_1 \mapsto 0$ (same argument)

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

$\overline{y}_1 \vee y_0$

$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$

$y_1 \vee \overline{y}_0 \vee \overline{x}_1$

Derivable by redundance with witness $\omega : y_1 \mapsto 1$

$$F \wedge D \wedge \neg(y_1 \vee \overline{y}_0 \vee \overline{x}_1) \models (F \wedge D)\restriction_\omega \wedge \{y_1 \vee \overline{y}_0 \vee \overline{x}_1\}\restriction_\omega$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

$$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$$

$$y_1 \vee \overline{y}_0 \vee \overline{x}_1$$

Derivable by redundance with witness $\omega : y_1 \mapsto 1$

$$F \wedge D \wedge \neg(y_1 \vee \overline{y}_0 \vee \overline{x}_1)$$
$$\models (F \wedge D)\restriction_\omega \wedge \{y_1 \vee \overline{y}_0 \vee \overline{x}_1\}\restriction_\omega$$
$$F \wedge D \wedge \{\overline{y}_1 \wedge y_0 \wedge x_1)$$
$$\models \cdots \wedge D\restriction_\omega \wedge \dots$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

$$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$$

$$y_1 \vee \overline{y}_0 \vee \overline{x}_1$$

Derivable by redundance with witness $\omega : y_1 \mapsto 1$

$$F \wedge D \wedge \neg(y_1 \vee \overline{y}_0 \vee \overline{x}_1)$$

$$\models (F \wedge D)\restriction_\omega \wedge \{y_1 \vee \overline{y}_0 \vee \overline{x}_1\}\restriction_\omega$$

$$F \wedge D \wedge \{\overline{y}_1 \wedge y_0 \wedge x_1)$$

$$\models \dots \wedge D\restriction_\omega \wedge \dots$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

$$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$$

$$y_1 \vee \overline{y}_0 \vee \overline{x}_1$$

Derivable by redundance with witness $\omega : y_1 \mapsto 1$

$$F \wedge D \wedge \neg(y_1 \vee \overline{y}_0 \vee \overline{x}_1)$$
$$\models (F \wedge D)\restriction_\omega \wedge \{y_1 \vee \overline{y}_0 \vee \overline{x}_1\}\restriction_\omega$$
$$F \wedge D \wedge \{\overline{y}_1 \wedge y_0 \wedge x_1)$$
$$\models \cdots \wedge D\restriction_\omega \wedge \ldots$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

$$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$$

$$y_1 \vee \overline{y}_0 \vee \overline{x}_1$$

$$y_1 \vee \overline{y}_0 \vee \sigma(x_1)$$

Derivable by redundance with witness $\omega : y_1 \mapsto 1$ (same argument)

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

$\overline{y}_1 \vee y_0$

$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$

$y_1 \vee \overline{y}_0 \vee \overline{x}_1$

$y_1 \vee \overline{y}_0 \vee \sigma(x_1)$

$\overline{y}_1 \vee \overline{x}_2 \vee \sigma(x_2)$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

y_0

$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$

$\overline{y}_1 \vee y_0$

$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$

$y_1 \vee \overline{y}_0 \vee \overline{x}_1$

$y_1 \vee \overline{y}_0 \vee \sigma(x_1)$

$\overline{y}_1 \vee \overline{x}_2 \vee \sigma(x_2)$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$+ 2^{n-1} \cdot \left(\overline{y}_1 + \overline{\sigma(x_1)} + x_1 \geq 1\right)$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \lor \overline{x}_1 \lor \sigma(x_1)$$

$$\overline{y}_1 \lor y_0$$

$$\overline{y}_1 \lor \overline{\sigma(x_1)} \lor x_1$$

$$y_1 \lor \overline{y}_0 \lor \overline{x}_1$$

$$y_1 \lor \overline{y}_0 \lor \sigma(x_1)$$

$$\overline{y}_1 \lor \overline{x}_2 \lor \sigma(x_2)$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$+\, 2^{n-1} \cdot \left(\overline{y}_1 + \overline{\sigma(x_1)} + x_1 \geq 1\right)$$

$$2^{n-1} \cdot \overline{y}_1 + \sum_{i=2}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

DETAILED DERIVATION OF CNF BREAKING CONSTRAINTS

Derived constraints (D):

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$y_0$$

$$\overline{y}_0 \vee \overline{x}_1 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee y_0$$

$$\overline{y}_1 \vee \overline{\sigma(x_1)} \vee x_1$$

$$y_1 \vee \overline{y}_0 \vee \overline{x}_1$$

$$y_1 \vee \overline{y}_0 \vee \sigma(x_1)$$

$$\overline{y}_1 \vee \overline{x}_2 \vee \sigma(x_2)$$

$$\sum_{i=1}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

$$+ 2^{n-1} \cdot \left(\overline{y}_1 + \overline{\sigma(x_1)} + x_1 \geq 1\right)$$

$$2^{n-1} \cdot \overline{y}_1 + \sum_{i=2}^{n} 2^{n-i} \cdot (\sigma(x_i) - x_i) \geq 0$$

The clause to derive is RUP wrt this constraint

REFERENCES

[ASM06] Fadi A. Aloul, Karem A. Sakallah, and Igor L. Markov. Efficient symmetry breaking for Boolean satisfiability. *IEEE Transactions on Computers*, 55(5):549–558, 2006.

[BGMN22] Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI '22)*, pages 3698–3707, February 2022.

[BKG19] Curtis Bright, Ilias S. Kotsireas, and Vijay Ganesh. SAT solvers and computer algebra systems: a powerful combination for mathematics. In Tima Pakfetrat, Guy-Vincent Jourdan, Kostas Kontogiannis, and Robert F. Enenkel, editors, *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, CASCON 2019, Markham, Ontario, Canada, November 4-6, 2019*, pages 323–328. ACM, 2019.

[BNOS10] Belaïd Benhamou, Tarek Nabhani, Richard Ostrowski, and Mohamed Réda Saïdi. Enhancing clause learning by symmetry in SAT solvers. In *Proceedings of the 2010 22Nd IEEE International Conference on Tools with Artificial Intelligence - Volume 01*, ICTAI '10, pages 329–335, Washington, DC, USA, 2010. IEEE Computer Society.

[Bre] Breakid. `https://bitbucket.org/krr/breakid`.

[BS07] Belaïd Benhamou and Mohamed Réda Saïdi. Dynamic detection and elimination of local symmetry in CSPs. 2007.

[BT19] Samuel R. Buss and Neil Thapen. DRAT proofs, propagation redundancy, and extended resolution. In *Proceedings of the 22nd International Conference on Theory and Applications of Satisfiability Testing (SAT '19)*, volume 11628 of *Lecture Notes in Computer Science*, pages 71–89. Springer, July 2019.

REFERENCES

[CCT87] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[DBB17] Jo Devriendt, Bart Bogaerts, and Maurice Bruynooghe. Symmetric explanation learning: Effective dynamic symmetry handling for SAT. In *Proceedings of the 20th International Conference on Theory and Applications of Satisfiability Testing (SAT '17)*, volume 10491 of *Lecture Notes in Computer Science*, pages 83–100. Springer, August 2017.

[DBBD16] Jo Devriendt, Bart Bogaerts, Maurice Bruynooghe, and Marc Denecker. Improved static symmetry breaking for SAT. In *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT '16)*, volume 9710 of *Lecture Notes in Computer Science*, pages 104–122. Springer, July 2016.

[DBD$^+$12] Jo Devriendt, Bart Bogaerts, Broes De Cat, Marc Denecker, and Christopher Mears. Symmetry propagation: Improved dynamic symmetry breaking in SAT. In *IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012*, pages 49–56. IEEE Computer Society, 2012.

[GMNO22] Stephan Gocht, Ruben Martins, Jakob Nordström, and Andy Oertel. Certified CNF translations for pseudo-Boolean solving. In *Proceedings of the 25th International Conference on Theory and Applications of Satisfiability Testing (SAT '22)*, volume 236 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:25, August 2022.

[GN21] Stephan Gocht and Jakob Nordström. Certifying parity reasoning efficiently using pseudo-Boolean proofs. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI '21)*, pages 3768–3777, February 2021.

[Goc22] Stephan Gocht. Certifying correctness for combinatorial algorithms by using pseudo-boolean reasoning, 2022.

REFERENCES

[HHW15] Marijn J. H. Heule, Warren A. Hunt Jr., and Nathan Wetzler. Expressing symmetry breaking in DRAT proofs. In *Proceedings of the 25th International Conference on Automated Deduction (CADE-25)*, volume 9195 of *Lecture Notes in Computer Science*, pages 591–606. Springer, August 2015.

[HKB17] Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Short proofs without new variables. In *Proceedings of the 26th International Conference on Automated Deduction (CADE-26)*, volume 10395 of *Lecture Notes in Computer Science*, pages 130–147. Springer, August 2017.

[HKM$^+$05] Marijn Heule, Alexander Keur, Hans Van Maaren, Coen Stevens, and Mark Voortman. CNF symmetry breaking options in conflict driven SAT solving, 2005.

[JHB12] Matti Järvisalo, Marijn J. H. Heule, and Armin Biere. Inprocessing rules. In *Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR '12)*, volume 7364 of *Lecture Notes in Computer Science*, pages 355–370. Springer, June 2012.

[KS21] Markus Kirchweger and Stefan Szeider. SAT modulo symmetries for graph generation. In Laurent D. Michel, editor, *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*, volume 210 of *LIPIcs*, pages 34:1–34:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[MBCK18] Hakan Metin, Souheib Baarir, Maximilien Colange, and Fabrice Kordon. Cdclsym: Introducing effective symmetry breaking in SAT solving. In *TACAS 2018, Proceedings, Part I*, pages 99–114, 2018.

[MBK19] Hakan Metin, Souheib Baarir, and Fabrice Kordon. Composing symmetry propagation and effective symmetry breaking for SAT solving. In *NASA Formal Methods, Proceedings*, pages 316–332, 2019.

REFERENCES

[Sab09] Ashish Sabharwal. SymChaff: Exploiting symmetry in a structure-aware satisfiability solver. *Constraints*, 14(4):478–505, 2009.

[SHvM09] Bas Schaafsma, Marijn Heule, and Hans van Maaren. Dynamic symmetry breaking by simulating Zykov contraction. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, volume 5584 of *Lecture Notes in Computer Science*, pages 223–236. Springer, 2009.

[TD20] Rodrigue Konan Tchinda and Clémentin Tayou Djamégni. On certifying the UNSAT result of dynamic symmetry-handling-based SAT solvers. *Constraints*, 25(3–4):251–279, December 2020.