On row symmetry in SAT

Jo Devriendt* Bart Bogaerts Maurice Bruynooghe
firstname.lastname@cs.kuleuven.be

KU Leuven - University of Leuven

July 18, 2014

CP perspective

- CSP: (V, D, C)
- assignment $\alpha : (V \rightarrow D)$
- solution satisfies constraints

CP perspective

- CSP: (V, D, C)
- assignment $\alpha : (V \to D)$
- solution satisfies constraints

For example:

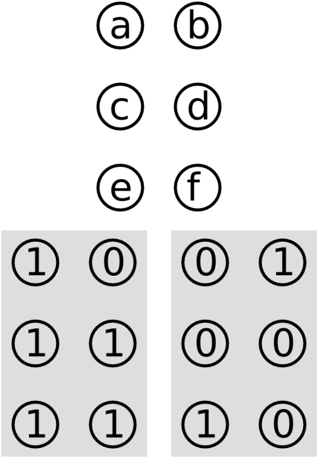
- $V = \{a, b, c, d, e, f\}$
- $D = \{0, 1\}$
- $C = \{b \leq 0\}$
- $\alpha = \{a = 1, b = 0, c = 1,$
  $d = 1, e = 1, f = 1\}$
- $\alpha' = \{a = 0, b = 1, c = 0,$
  $d = 0, e = 1, f = 0\}$

CP perspective

- CSP: (V, D, C)
- assignment $\alpha : (V \rightarrow D)$
- solution satisfies constraints

For example:

- $V = \{a, b, c, d, e, f\}$
- $D = \{0, 1\}$
- $C = \{b \leq 0\}$
- $\alpha = \{a = 1, b = 0, c = 1, d = 1, e = 1, f = 1\}$
- $\alpha' = \{a = 0, b = 1, c = 0, d = 0, e = 1, f = 0\}$

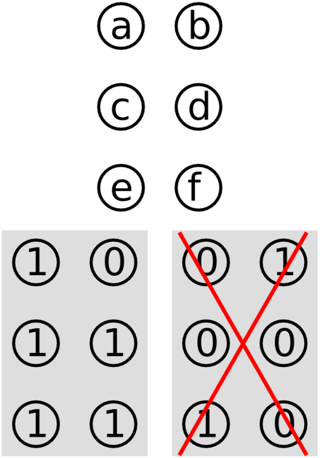


CP perspective

- CSP: (V, D, C)
- assignment $\alpha : (V \to D)$
- solution satisfies constraints

For example:

- $V = \{a, b, c, d, e, f\}$
- $D = \{0, 1\}$
- $C = \{b \leq 0\}$
- $\alpha = \{a = 1, b = 0, c = 1, d = 1, e = 1, f = 1\}$
- $\alpha' = \{a = 0, b = 1, c = 0, d = 0, e = 1, f = 0\}$



CP perspective on symmetry

$S : (V \to D) \to (V \to D)$
symmetry S is a permutation of the set of assignments preserving satisfaction to all constraints
Recall:

- $V = \{a, b, c, d, e, f\}$
- $D = \{0, 1\}$
- $C = \{b \leq 0\}$

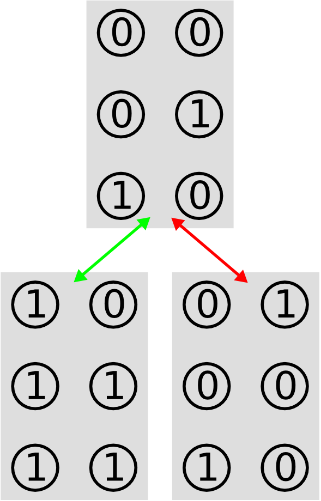
CP perspective on symmetry

$S : (V \to D) \to (V \to D)$

symmetry S is a permutation of the set of assignments preserving satisfaction to all constraints

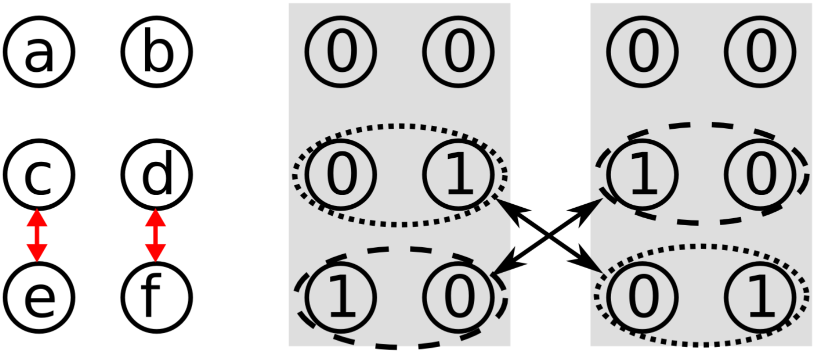
Recall:

- $V = \{a, b, c, d, e, f\}$
- $D = \{0, 1\}$
- $C = \{b \leq 0\}$



CP perspective on symmetry

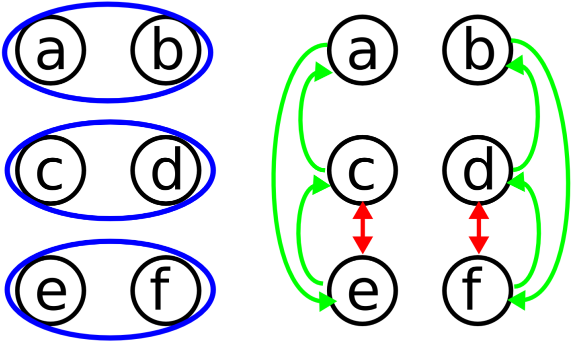
Common restriction:
variable symmetry $S_P : \alpha \mapsto \alpha \circ P$
induced by variable permutation $P : V \to V$
For example: $P = (ce)(df)$



CP perspective on row symmetry

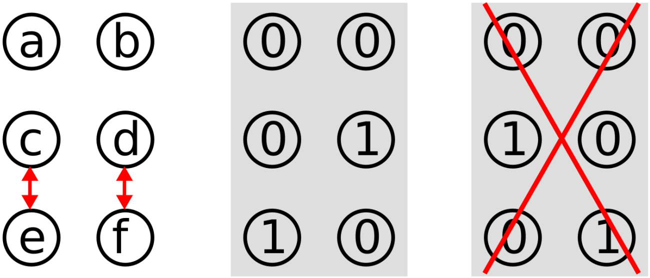
Row symmetry:

- special case of variable symmetry
- assumption: V is ordered as a matrix M
- P permutes the rows of M
- CSP is row-interchangeable for M iff all permutations on the rows of M induce a symmetry



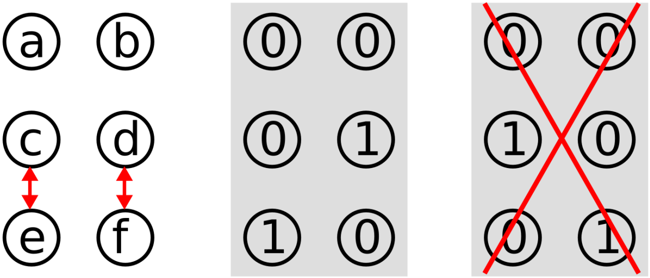
CP perspective on row symmetry

Symmetry breaking: speed up search by adding extra constraints to remove symmetrical assignments from the assignment space.



CP perspective on row symmetry

Symmetry breaking: speed up search by adding extra constraints to remove symmetrical assignments from the assignment space.

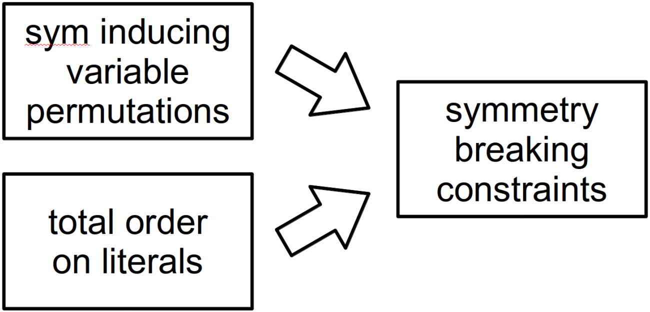


Complete symmetry breaking: maximal number of symmetric assignments removed while retaining soundness.

CP result: row interchangeability can be broken completely by enforcing lexicographic order on rows. [Flener e.a. 2001]

SAT perspective on breaking symmetry

General symmetry breaking in SAT as per Saucy+Shatter [Aloul e.a. 2006]:

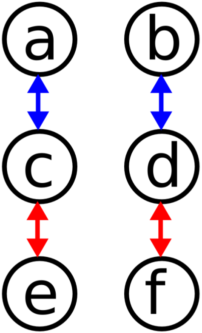


For each variable permutation P and for each variable v, add the lex-leader constraint $\forall v' \prec v : v' = P(v') \Rightarrow v \leq P(v)$.

Can Shatter completely break row interchangeability?

- variable permutations: P_{12}, P_{23} that induce row interchangeability
- order on variables: $a \prec b \prec c \prec d \prec e \prec f$
- symmetry breaking constraints: $a \leq c$, $a = c \Rightarrow b \leq d$, $c \leq e$, $c = e \Rightarrow d \leq f$

These constraints actually state that the rows of each solution must be lexicographically ordered!

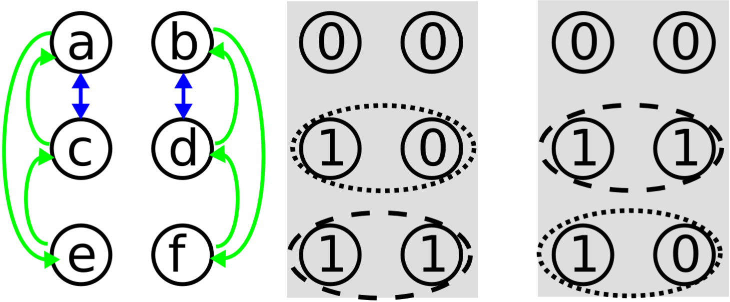


Row interchangeability can be completely broken by standard SAT symmetry breaking methods, given the right variable permutations and variable ordering.

What can go wrong?

- variable permutations: P_{12}, P_{321}
  that induce row interchangeability
- order on variables: $a \prec b \prec c \prec d \prec e \prec f$
- symmetry breaking constraints:
  $a \leq c$, $a = c \Rightarrow b \leq d$,
  $a \leq e$, $a = e \Rightarrow b \leq f$,
  $a = e \wedge b = f \Rightarrow c \leq a$,$a = e \wedge b = f \wedge c = a \Rightarrow d \leq b$

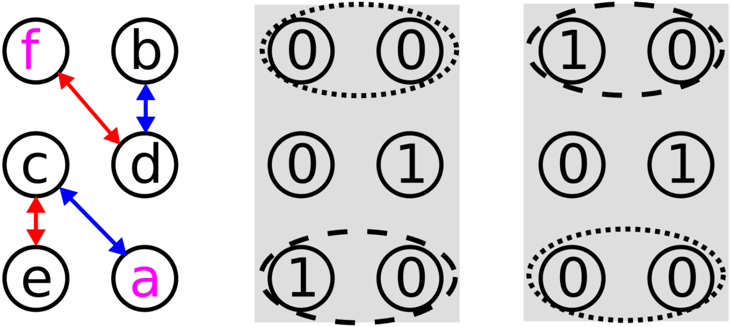
These do not represent lexicographic row ordering constraint



What can go wrong?

- variable permutations: P_{12}, P_{23} that induce row interchangeability
- order on variables: $f \prec b \prec c \prec d \prec e \prec a$
- symmetry breaking constraints:
  $b \leq d$, $b = d \Rightarrow c \leq a$,
  $f \leq d$, $f = d \Rightarrow c \leq e$

These do not represent lexicographic row ordering constraint



SAT perspective: conclusion

To completely break row interchangeability:

- need for right generator symmetries
- need for right ordering on variables

Easy when you know the matrix structure of the variables...

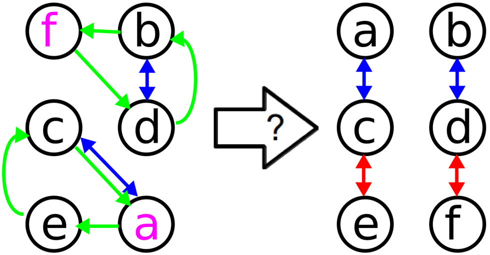
How to detect matrix structure of variables in a CNF?
How to detect row interchangeability in a CNF?

Row Interchangeability Detection

Problem statement:

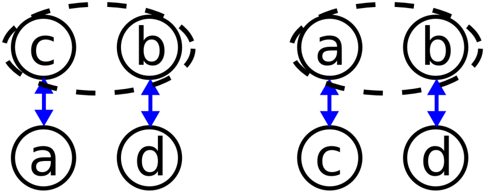
Row Interchangeability Detection (RID)
Given a CNF, find a maximal set of variables that form a variable matrix with interchangeable rows.

Chosen problem perspective: start from detected symmetry group.



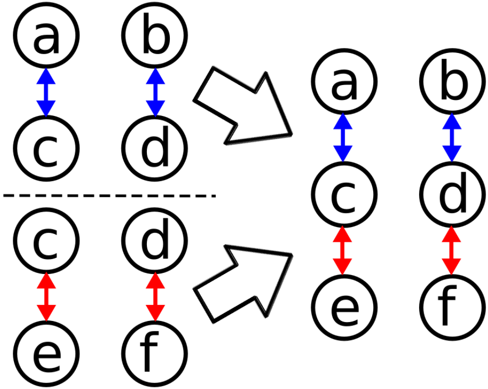
Involution symmetries form rows

A permutation P for which $P^2 = I$, is an involution. Each variable involution forms multiple interchangeable row matrices, with only 2 rows:



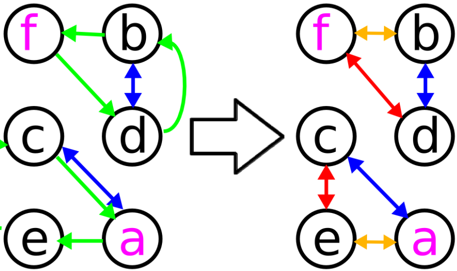
Involution symmetries form rows

Compatible involutions can be combined to row interchangeable matrices with more than 2 rows:



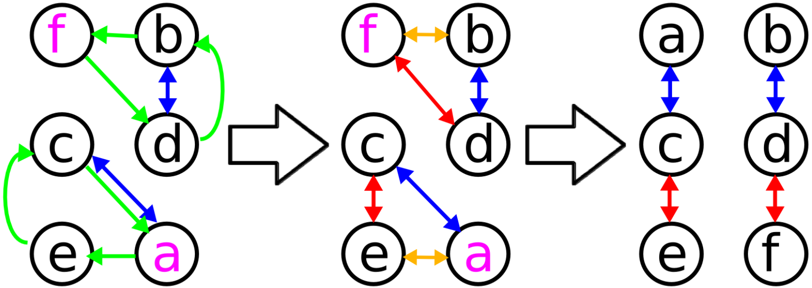
Heuristically search for involutions

Start from generator symmetries returned by Saucy. Compose symmetries to form *small* involutions.



Matrix structure detection by involutions

Combining involution generation & matrix extraction solves RID:



Can be extended to the piecewise case, where multiple disjoint row interchangeable variable matrices exist for one problem.

Piecewise Row Interchangeability Detection – the **PRID** problem.

Working implementation: BreakIDGlucose

- extends Shatter with row interchangeability handling
  - use Saucy to detect symmetry inducing variable permutations
  - new: generate involutions to solve PRID
  - new: add row involutions to set of variable permutations
  - new: adjust order on variables as per detected rows
  - add Shatter's lex-leader constraints
- used Glucose 2.1 as SAT solving engine
- obtained gold medal at 2013's SAT competition hard combinatorial track
- could only solve 5 out of 14 two pigeon hole instances

Complete row interchangeability breaking is relevant in SAT
Improvement possible with better detection of matrix structure – better answer to PRID.

Conclusion

Our contribution:

- We showed how complete row interchangeability symmetry breaking can be incorporated in existing SAT symmetry breaking methods
- We designed a first PRID solving attempt that reasons on involution symmetries
- We implemented these ideas in the award-winning solver BreakIDGlucose

Future work:

- Better answer to the PRID problem, perhaps by adjusting the symmetry detection algorithms?

Where does row interchangeability in SAT come from?

SAT encodings of

- variable interchangeable CSP's
- value interchangeable CSP's
- row interchangeable CSP's
- relational model generation problems where relation $R : D_1 \times \ldots \times D_n$ has to be found and some disjoint D_i contains interchangeable elements.
  For example: the IDP system.

Note the triviality of solving the PRID problem in such a high level language!
Analogous to cardinality constraint problem in SAT.

Thanks for your attention!
Questions?

Flener e.a. 2001: Symmetry in Matrix Models
Aloul e.a. 2006: Efficient symmetry breaking for Boolean satisfiability