

Exploitation des symétries dynamiques pour la résolution des problèmes SAT

Thèse de doctorat de Sorbonne Université

Hakan METIN

Jury Members:

PASCAL FONTAINE

LAURE PETRUCCI

JEAN-MICHEL COUVREUR

EMANUELLE ENCRENAZ

SOUHEIB BAARIR

FABRICE KORDON

Maître de conférences, Université de Liège

Professeur, Université Paris 13

Professeur, Université d'Orléans

Maître de conférences, Sorbonne Université

Maître de conférences, Université Paris Nanterre

Professeur, Sorbonne Université

Supervisors:

SOUHEIB BAARIR

FABRICE KORDON

Maître de conférences, Université Paris Nanterre

Professeur, Sorbonne Université



Motivation

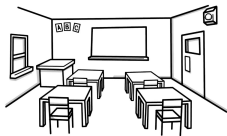
SAT is widely used in different domains:

- Artificial intelligence (planning, games, ...)
- Bioinformatics (haplotype inference, ...)
- Security (cryptanalysis, inversion attack on hash function)
- Computationally hard problems (graph coloring, ...)
- Formal Methods (hardware model checking, ...)

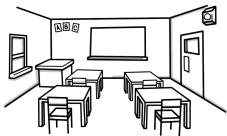
Outline

- 1 SAT overview
 - SAT basics
 - SAT and symmetries
- 2 Existing approaches
- 3 Contribution and results

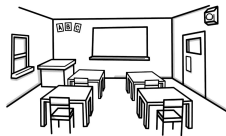
SAT an example



1



2



3



A



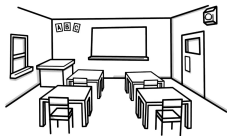
B



C

Is it possible to attribute each group to a classroom?

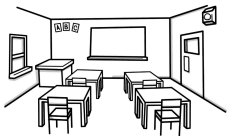
SAT an example



1
↑



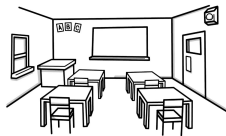
A



2
↑



B



3
↑

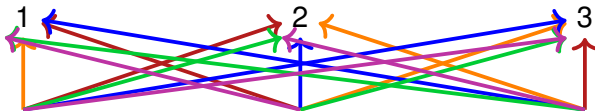
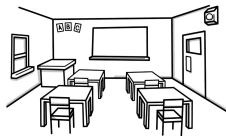
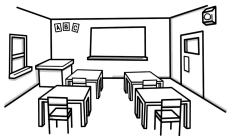
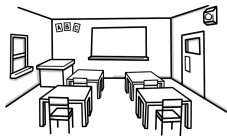


C

Is it possible to attribute each group to a classroom?

YES!

SAT an example



A



B

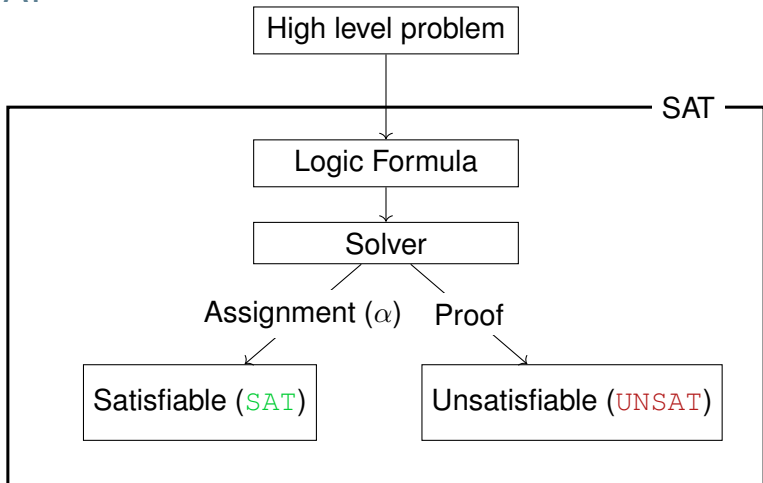


C

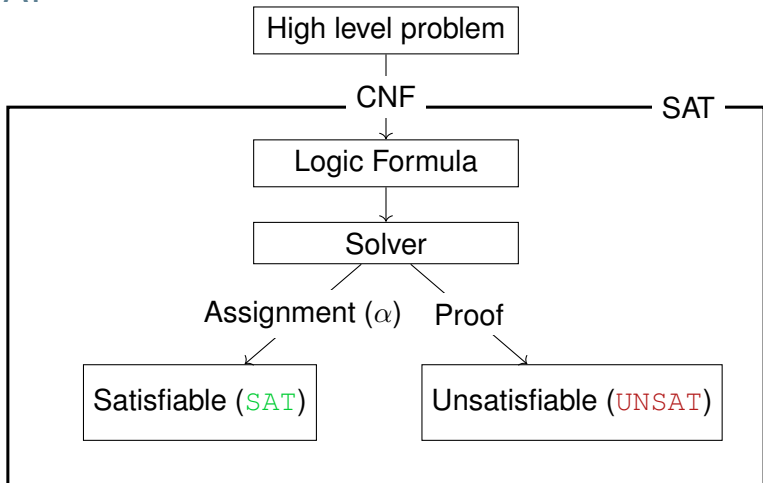
Is it possible to attribute each group to a classroom?

YES! Many solutions

SAT



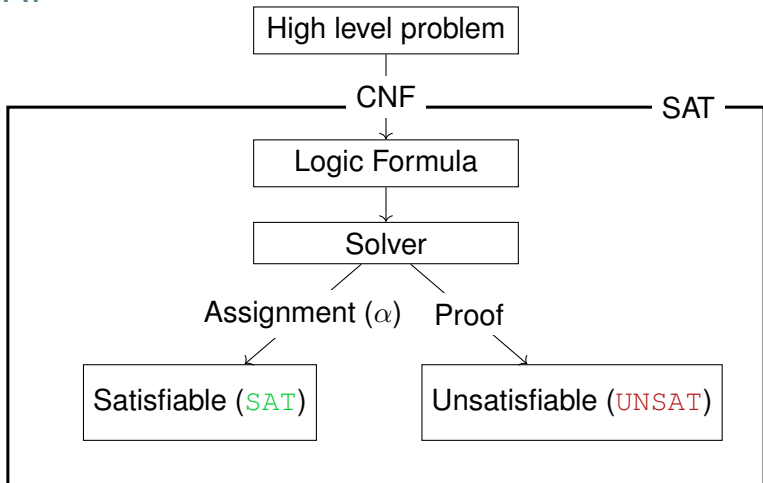
SAT



CNF Representation:

$$\underbrace{(x_1 \vee x_2 \vee \neg x_3)}_{\text{Clause with literals } x_1, x_2, \neg x_3}$$

SAT

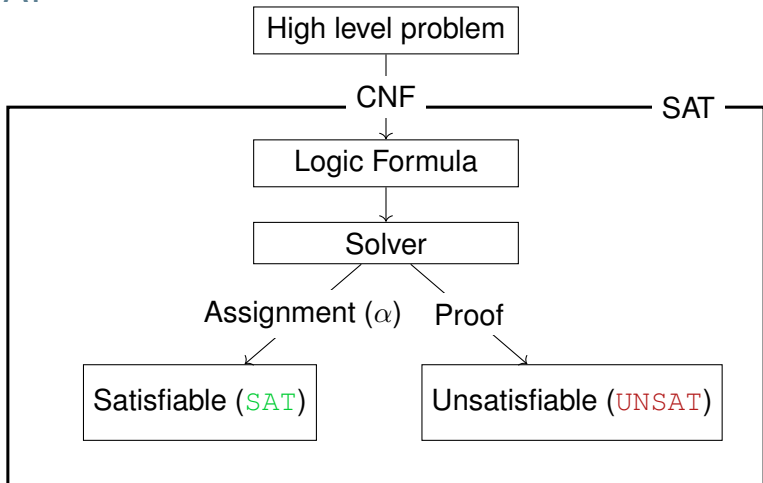


CNF Representation:

$$\underbrace{(x_1 \vee x_2 \vee \neg x_3)}_{\text{Clause}} \wedge (\neg x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_4)$$

Formula (CNF)

SAT



CNF Representation:

$$(x_1 \vee x_2 \vee \neg x_3) \rightarrow \{x_1, x_2, \neg x_3\}$$

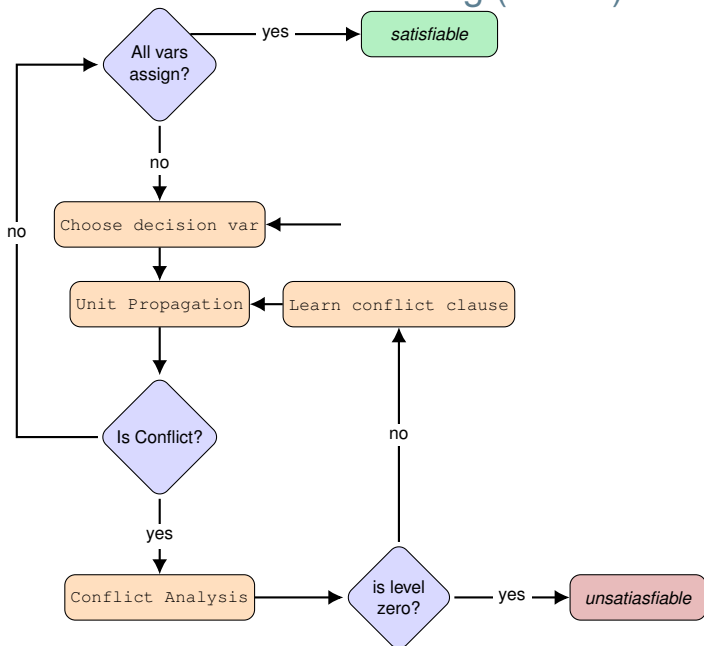
SAT Solving

Solving SAT formula is known to be **NP-complete** [Coo71]

Enumerative Algorithm:

- Davis, Putnam, Logemann, and Loveland (DPLL) [DLL62]
 - Boolean Constraint Propagation (BCP)
- Conflict Driven Clause Learning (CDCL) [MSS99]
 - derived from DPLL
 - clause learning

Conflict Driven Clause Learning (CDCL)



CDCL in action TODO



$$\omega_1 = \{x_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, x_6\}$$

$$\omega_3 = \{\neg x_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$\omega_1 = \{\mathbf{x}_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, x_6\}$$

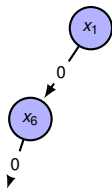
$$\omega_3 = \{\neg \mathbf{x}_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$\omega_1 = \{\mathbf{x}_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, \mathbf{x}_6\}$$

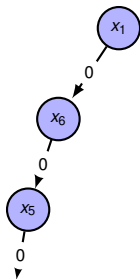
$$\omega_3 = \{\neg \mathbf{x}_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg \mathbf{x}_6\}$$

CDCL in action TODO



$$\omega_1 = \{\mathbf{x}_1, x_2, x_3\}$$

$$\omega_2 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$$

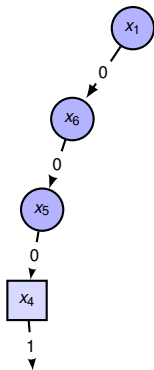
$$\omega_3 = \{\neg x_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg \mathbf{x}_6\}$$

CDCL in action TODO



$$\omega_1 = \{\mathbf{x}_1, x_2, x_3\}$$

$$\omega_2 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$$

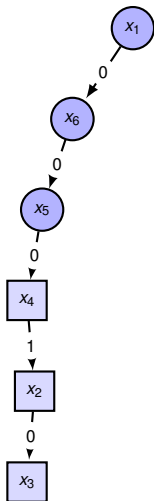
$$\omega_3 = \{\neg x_1, \neg x_5\}$$

$$\omega_4 = \{\neg \mathbf{x}_2, \neg \mathbf{x}_4\}$$

$$\omega_5 = \{\neg \mathbf{x}_3, \neg \mathbf{x}_4\}$$

$$\omega_6 = \{\neg x_3, \neg \mathbf{x}_6\}$$

CDCL in action TODO



$$\omega_1 = \{x_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, x_6\}$$

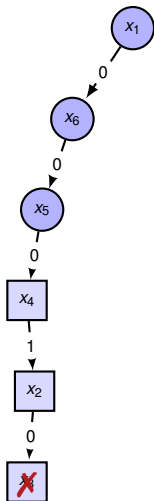
$$\omega_3 = \{\neg x_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$\omega_1 = \{x_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, x_6\}$$

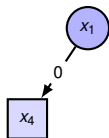
$$\omega_3 = \{\neg x_1, \neg x_5\}$$

$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$\omega_1 = \{x_1, x_2, x_3\}$$

$$\omega_2 = \{x_4, x_5, x_6\}$$

$$\omega_3 = \{\neg x_1, \neg x_5\}$$

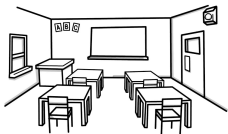
$$\omega_4 = \{\neg x_2, \neg x_4\}$$

$$\omega_5 = \{\neg x_3, \neg x_4\}$$

$$\omega_6 = \{\neg x_3, \neg x_6\}$$

$$\omega_7 = \{x_1, \neg x_4\}$$

An UNSAT example



1



2



A



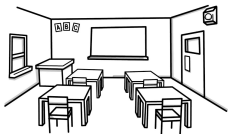
B



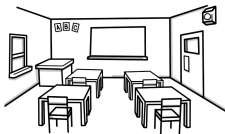
C

Is it possible to attribute each group to a classroom?

An UNSAT example



1



2



A



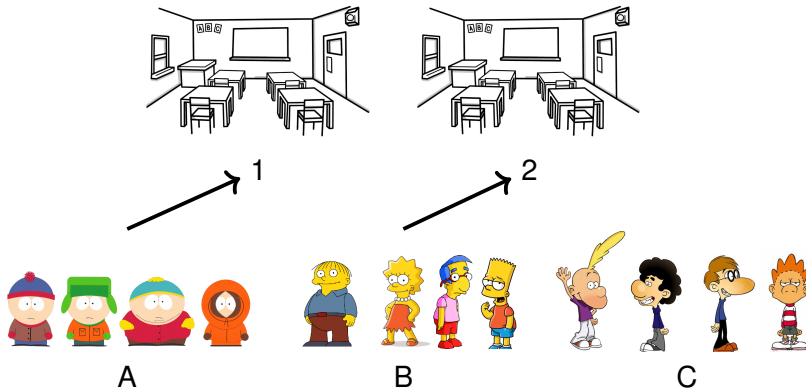
B



C

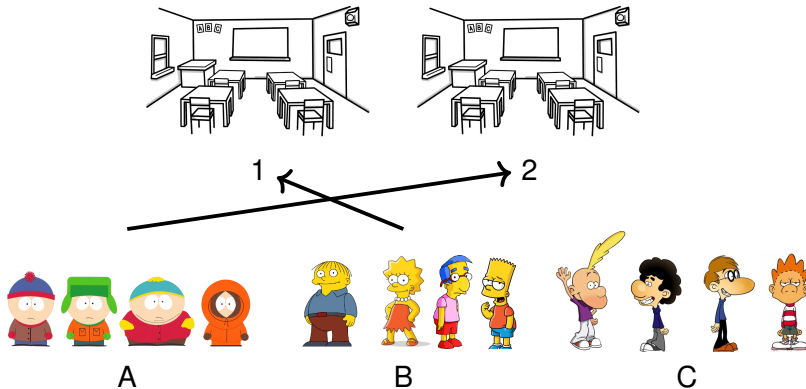
Is it possible to attribute each group to a classroom?

An UNSAT example



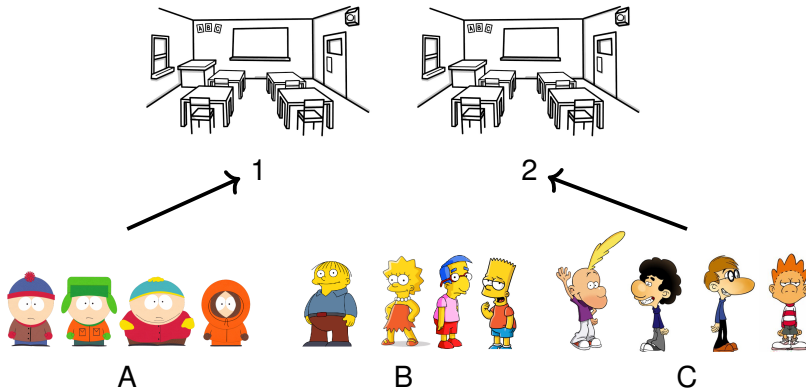
Is it possible to attribute each group to a classroom?

An UNSAT example



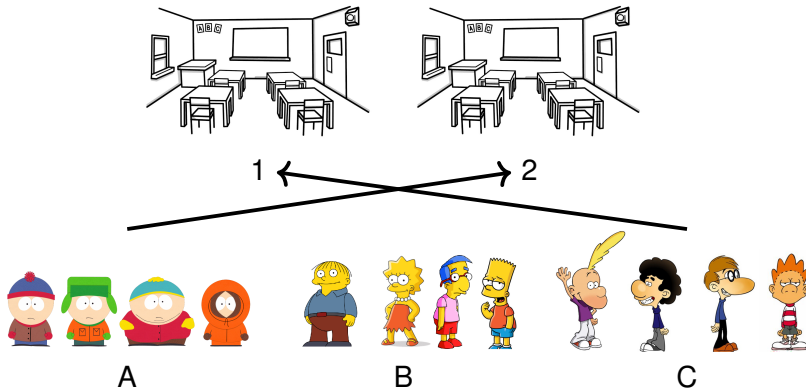
Is it possible to attribute each group to a classroom?

An UNSAT example



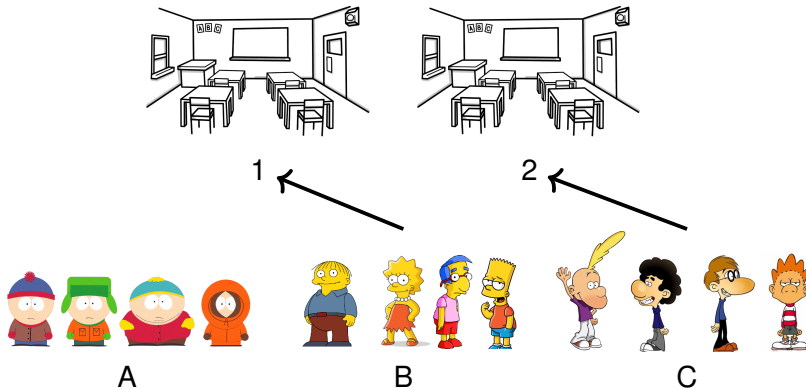
Is it possible to attribute each group to a classroom?

An UNSAT example



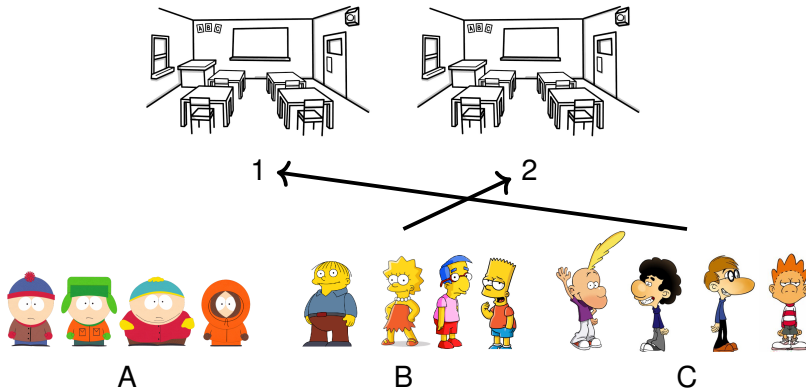
Is it possible to attribute each group to a classroom?

An UNSAT example



Is it possible to attribute each group to a classroom?

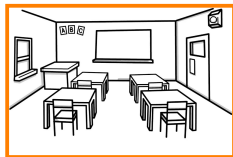
An UNSAT example



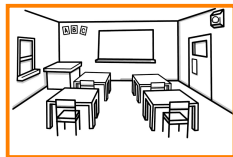
Is it possible to attribute each group to a classroom?

No!

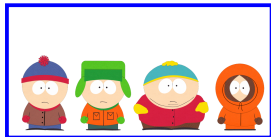
An UNSAT example



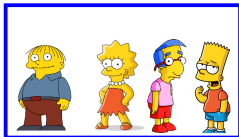
1



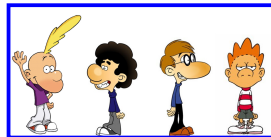
2



A



B



C

Is it possible to attribute each group to a classroom?

No!

Presence of symmetries hinders the performance of the solver

Outline

① SAT overview

- SAT basics

- SAT and symmetries

② Existing approaches

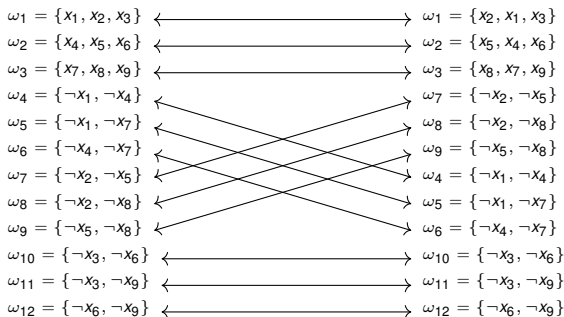
- Static symmetry breaking

- Dynamic symmetry breaking

③ Contribution and results

Symmetry

$$g = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\ x_2 & x_1 & x_3 & x_5 & x_4 & x_6 & x_8 & x_7 & x_9 \end{pmatrix} \rightarrow (x_1 \ x_2)(x_4 \ x_5)(x_7 \ x_8)$$



A symmetry (permutation) g is a bijective function (on variables) that leaves φ invariant.

Computing symmetries of a SAT problem

CNF formula

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \\ & \wedge (\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7) \\ & \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8) \\ & \wedge (\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9) \end{aligned}$$

¹<http://www.tcs.hut.fi/Software/bliss/>

²<http://vlsicad.eecs.umich.edu/BK/SAUCY/>

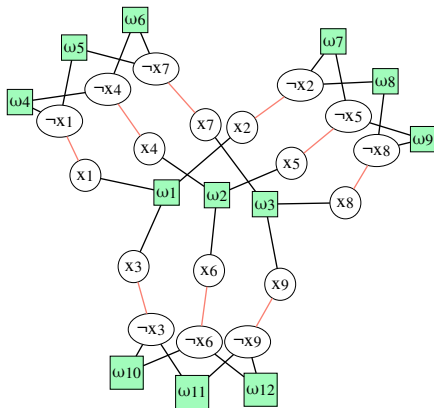
Computing symmetries of a SAT problem

CNF formula

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \\ & \wedge (\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7) \\ & \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8) \\ & \wedge (\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9) \end{aligned}$$



colored graph



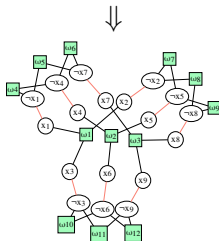
Computing symmetries of a SAT problem

CNF formula

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \\ & \wedge (\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7) \\ & \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8) \\ & \wedge (\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9) \end{aligned}$$



colored graph



graph automorphism



(bliss¹ or saucy²)

¹<http://www.tcs.hut.fi/Software/bliss/>

²<http://vlsicad.eecs.umich.edu/BK/SAUCY/>

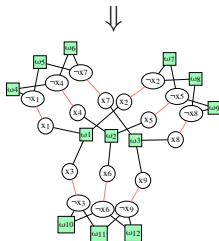
Computing symmetries of a SAT problem

CNF formula

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \\ & \wedge (\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7) \\ & \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8) \\ & \wedge (\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9) \end{aligned}$$



colored graph



⇓
graph automorphism

⇓
set of symmetries

⇓
(bliss¹ or saucy²)

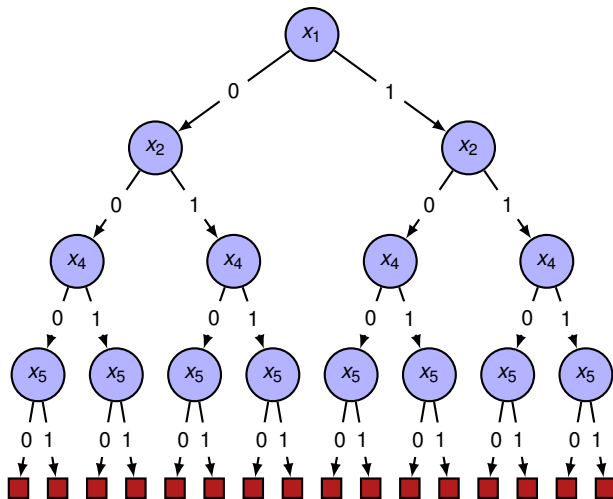
⇓

$$\begin{aligned} g_1 &= (x_2 \ x_3)(x_5 \ x_6)(x_8 \ x_9) \\ g_2 &= (x_4 \ x_7)(x_5 \ x_8)(x_6 \ x_9) \\ g_3 &= (x_1 \ x_2)(x_4 \ x_5)(x_7 \ x_8) \\ g_4 &= (x_1 \ x_4)(x_2 \ x_5)(x_3 \ x_6) \end{aligned}$$

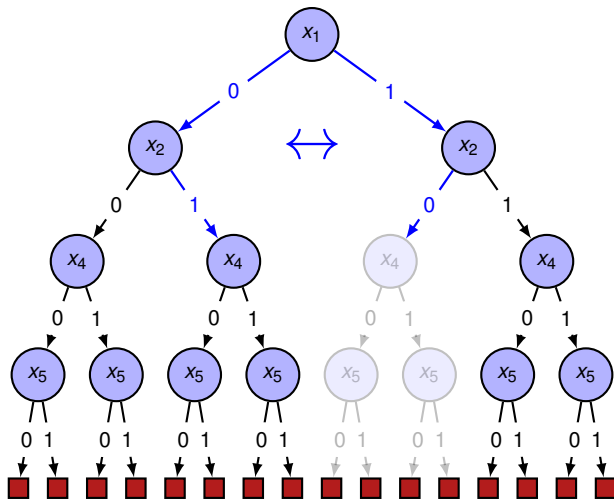
¹<http://www.tcs.hut.fi/Software/bliss/>

²<http://vlsicad.eecs.umich.edu/BK/SAUCY/>

Using symmetries to prune search space



Using symmetries to prune search space



Adds additional constraints to prune search space.

Generates symmetry breaking predicates (SBP)

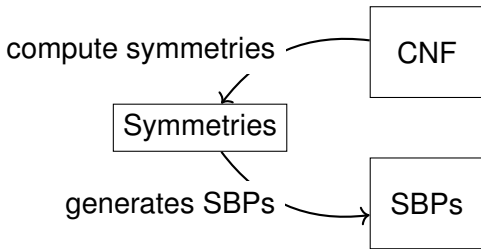
- Define lexicographic order
 - Define total order on variables
 - Define minimal value
- Forbid non minimal assignment with addition of SBP

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \quad \text{true} < \text{false}$$
$$(x_1 \ x_2)(x_4 \ x_5)(x_7 \ x_8)$$

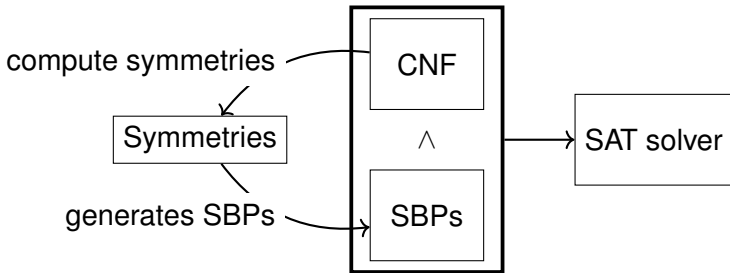
$x_1 \leq x_2$	$x_1 \vee \neg x_2$
$x_1 = x_2 \rightarrow x_4 \leq x_5$	$x_1 \vee x_2 \vee x_4 \vee \neg x_5$
	$\neg x_1 \vee \neg x_2 \vee x_4 \vee \neg x_5$
$x_1 = x_2 \wedge x_4 = x_5 \rightarrow x_8 \leq x_3$	$x_1 \vee x_2 \vee x_4 \vee x_5 \vee x_7 \vee \neg x_8$
	$\neg x_1 \vee \neg x_2 \vee x_4 \vee x_5 \vee x_7 \vee \neg x_8$
	...

Static symmetry breaking



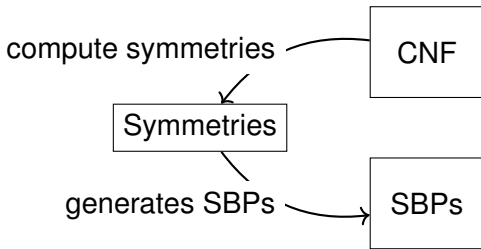
- Works well on many symmetric instances
- The solver can "explode" instead of being helped

Static symmetry breaking



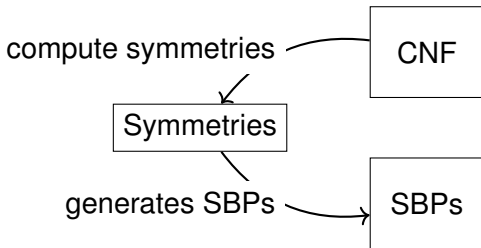
- Works well on many symmetric instances
- The solver can "explode" instead of being helped

Static symmetry breaking



- Works well on many symmetric instances
- The solver can "explode" instead of being helped

Static symmetry breaking



- Works well on many symmetric instances
- The solver can "explode" instead of being helped

Dynamic Symmetry Breaking

TODO

Accelerate tree traversal

modify solver

Different tools SP, SLS, SEL, ...

Symmetry Propagation

TODO

Present SP

Example

TODO

Build an example

Outline

① SAT overview

- SAT basics

- SAT and symmetries

② Existing approaches

- Static symmetry breaking

- Dynamic symmetry breaking

③ Contribution and results

- CDCL [Sym]

- Combination of different approaches

CDCL[Sym] idea

TODO

Tackling the explosion problem in the static symmetry breaking approaches.

Compute and inject ESBP opportunistically during the solving
Symmetry Controller in CDCL

Symmetry status

TODO

Reducer, Inactive, active

Example

TODO

Experimental results

TODO

ESBP + SP

TODO

Symmetry propagation on top of ESBP

Compose both approaches

Is it possible?

Notion of local symmetries

TODO

Computation of local symmetries

TODO

Experimental results

TODO

Conclusion and Perspective

TODO

Conclusion:

Perspectives:

Thanks !



Stephen A Cook.

The complexity of theorem-proving procedures.

In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.



Martin Davis, George Logemann, and Donald Loveland.

A machine program for theorem-proving.

Commun. ACM, 5(7):394–397, July 1962.



Joao P Marques-Silva and Karem A Sakallah.

Grasp: A search algorithm for propositional satisfiability.

IEEE Transactions on Computers, 48(5):506–521, 1999.