Exploitation des symétries dynamiques pour la résolution des problèmes SAT

Thèse de doctorat de Sorbonne Université

Hakan METIN

Supervisors:

SOUHEIB BAARIR	Maître de conférences, Université Paris Nanterre
FABRICE KORDON	Professeur, Sorbonne Université

Jury Members:

PASCAL FONTAINE	Maître de conférences, Université de Lorraine
LAURE PETRUCCI	Professeur, Université Paris 13
JEAN-MICHEL COUVREUR	Professeur, Université d'Orléans
EMANUELLE ENCRENAZ	Maître de conférences, Sorbonne Université
SOUHEIB BAARIR	Maître de conférences, Université Paris Nanterre
FABRICE KORDON	Professeur, Sorbonne Université

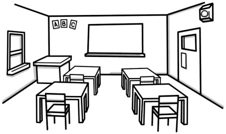
Motivation

SAT is widely used in different domains:

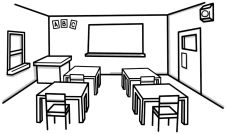
- Artificial intelligence (planning, games, ...)
- Bioinformatics (haplotype inference, ...)
- Security (cryptanalysis, inversion attack on hash function)
- Computationally hard problems (graph coloring, ...)
- Formal Methods (hardware model checking, ...)

Outline

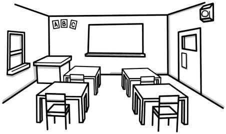
SAT an example



1

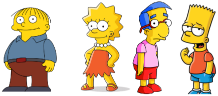


2



3

A

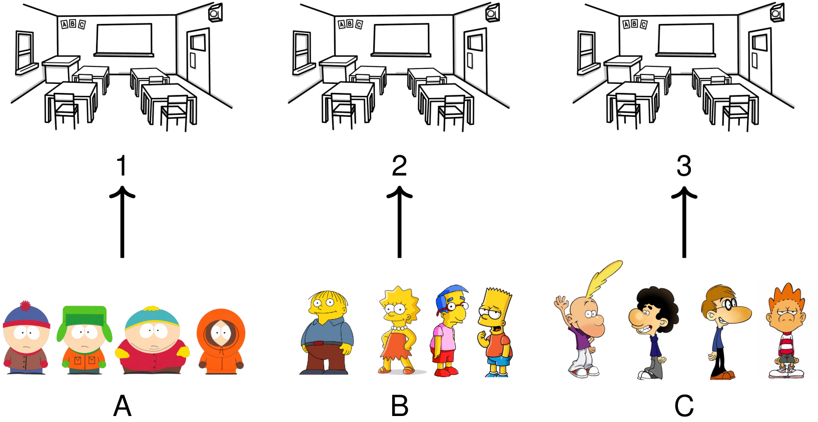


B

C

Is it possible to attribute each group to a classroom?

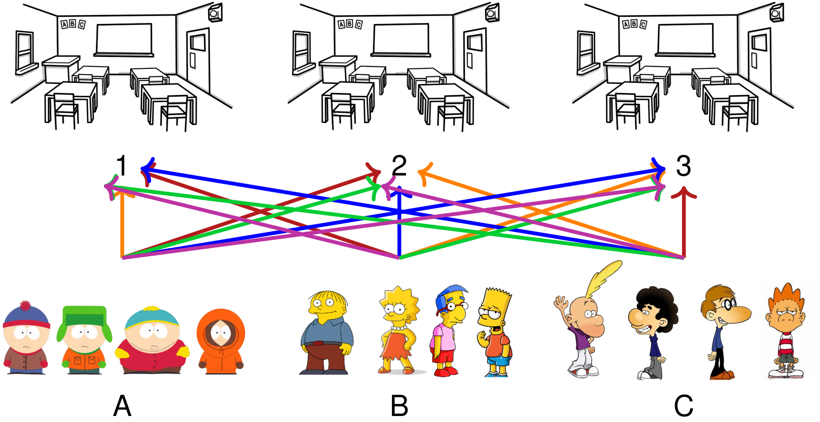
SAT an example



Is it possible to attribute each group to a classroom?

YES!

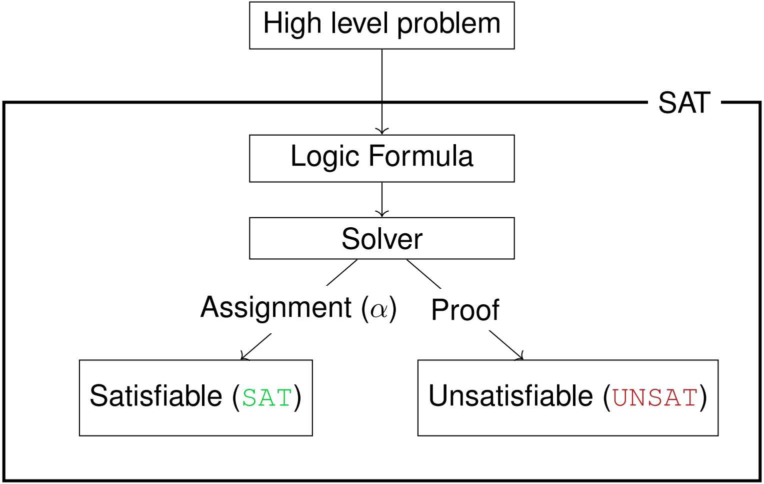
SAT an example



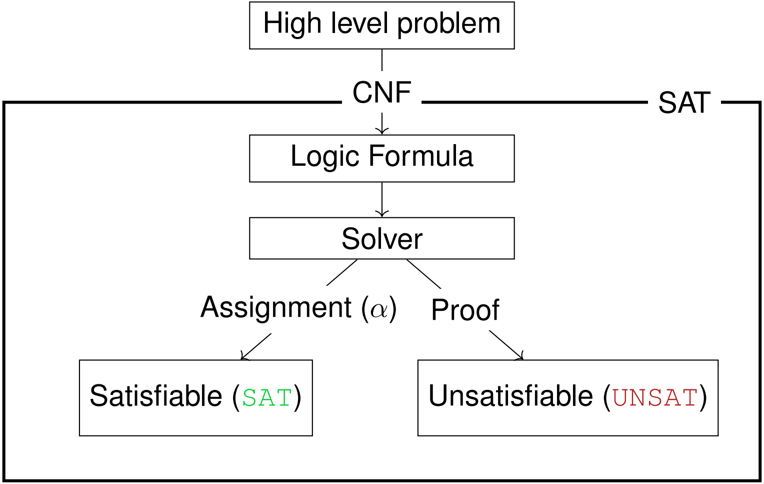
Is it possible to attribute each group to a classroom?

YES! Many solutions

SAT


High level problem
SAT
Logic Formula
Solver
Assignment (α)
Proof
Satisfiable (SAT)
Unsatisfiable (UNSAT)

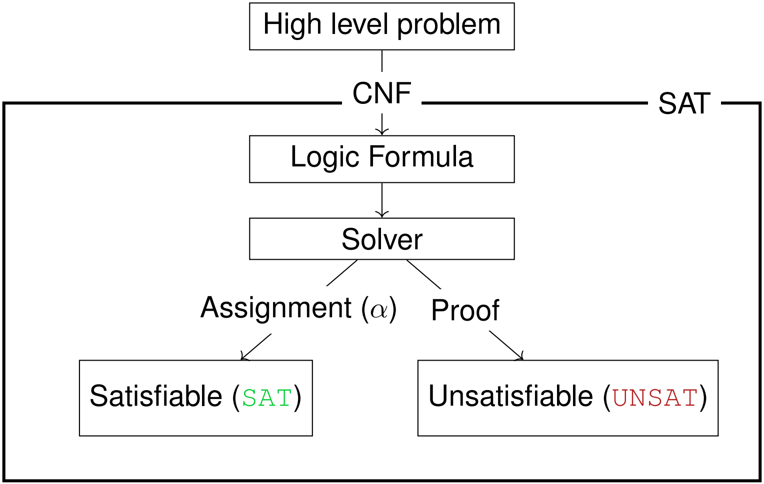
SAT



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:

$$\overbrace{\underbrace{(x_1 \lor x_2 \lor \neg x_3)}_{\textit{Clause}} \land (\neg x_1 \lor \neg x_2) \land (x_2 \lor \neg x_4)}^{\text{Formula (CNF)}}$$

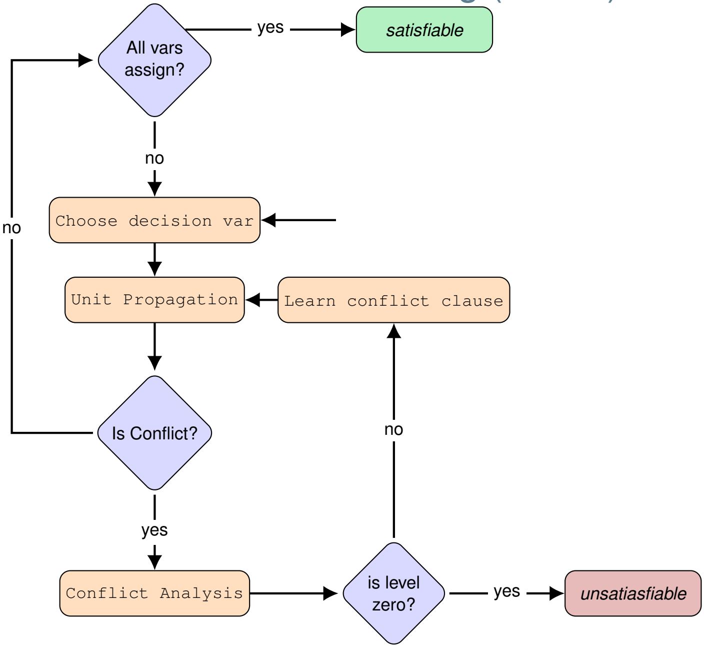
SAT Solving

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

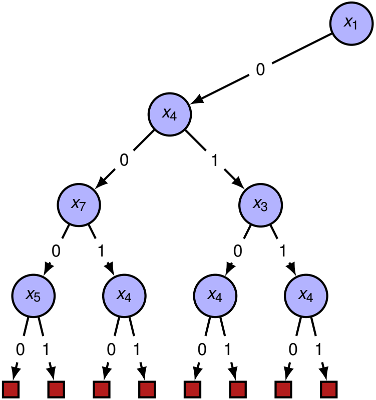
Enumerative Algorithm:

- 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)

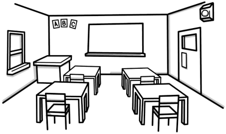


CDCL in action TODO

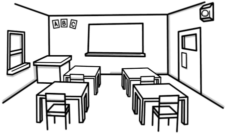


$$
\begin{aligned}
\omega_1 &= \{x_1, x_2, x_3\} \\
\omega_2 &= \{x_4, x_5, x_6\} \\
\omega_3 &= \{x_7, x_8, x_9\} \\
\omega_4 &= \{\neg x_1, \neg x_4\} \\
\omega_5 &= \{\neg x_1, \neg x_7\} \\
\omega_6 &= \{\neg x_4, \neg x_7\} \\
\omega_7 &= \{\neg x_2, \neg x_5\} \\
\omega_8 &= \{\neg x_2, \neg x_8\} \\
\omega_9 &= \{\neg x_5, \neg x_8\} \\
\omega_{10} &= \{\neg x_3, \neg x_6\} \\
\omega_{11} &= \{\neg x_3, \neg x_9\} \\
\omega_{12} &= \{\neg x_6, \neg x_9)
\end{aligned}
$$

An UNSAT example

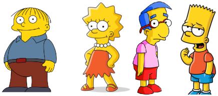


1



2

A

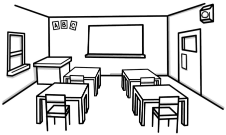


B

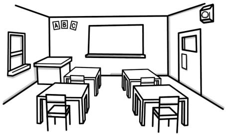
C

Is it possible to attribute each group to a classroom?

An UNSAT example

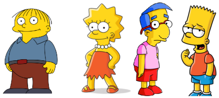


1

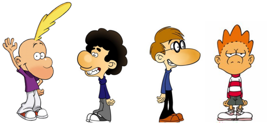


2

A



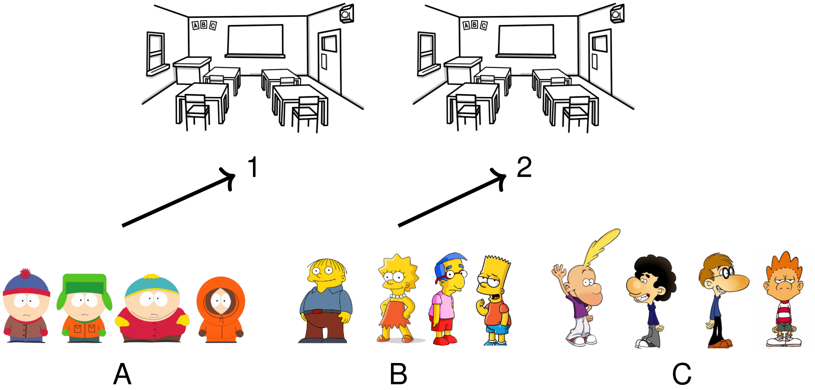
B



C

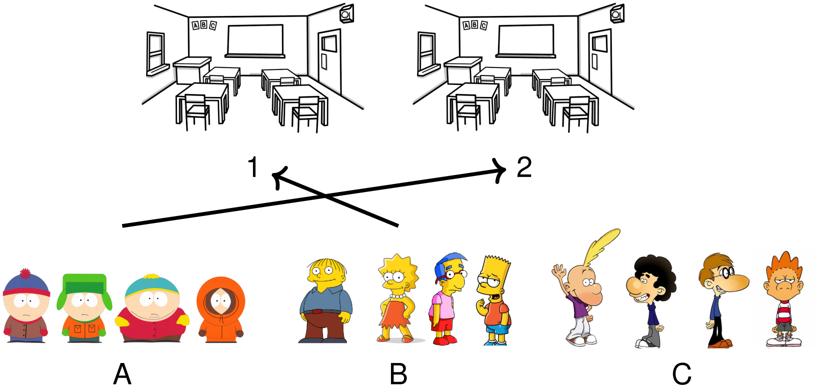
Is it possible to attribute each group to a classroom?

An UNSAT example



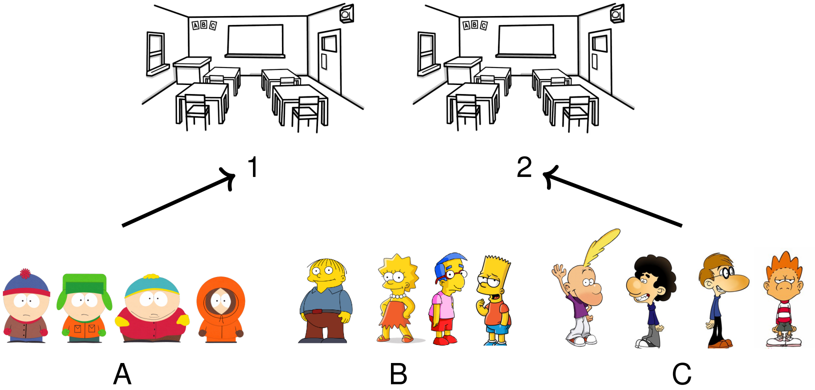
Is it possible to attribute each group to a classroom?

An UNSAT example



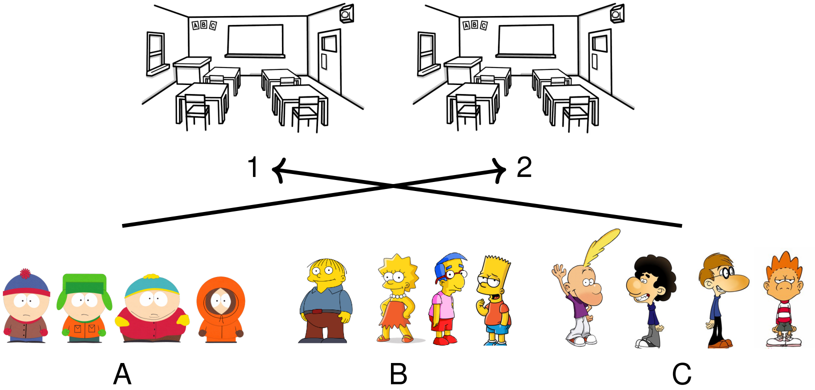
Is it possible to attribute each group to a classroom?

An UNSAT example



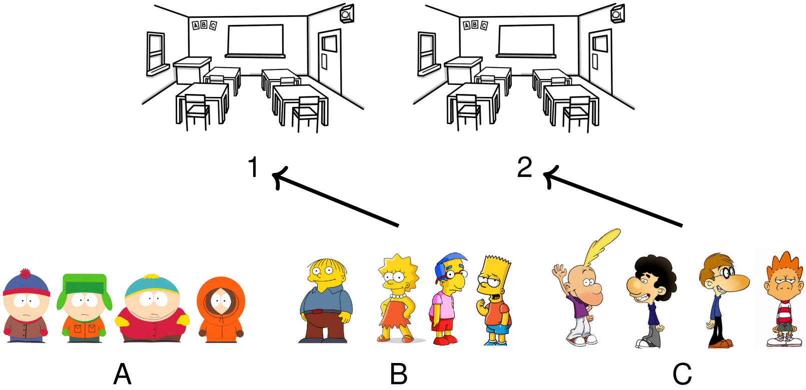
Is it possible to attribute each group to a classroom?

An UNSAT example



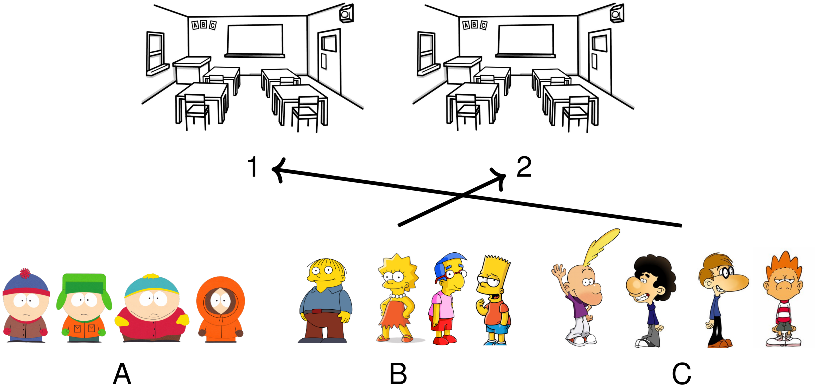
Is it possible to attribute each group to a classroom?

An UNSAT example



Is it possible to attribute each group to a classroom?

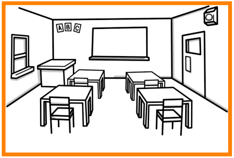
An UNSAT example



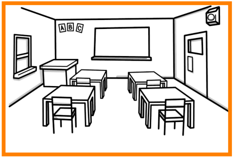
Is it possible to attribute each group to a classroom?

No!

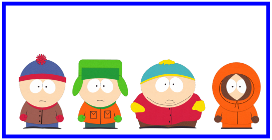
An UNSAT example



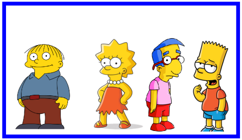
1



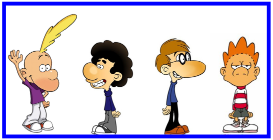
2



A



B



C

Is it possible to attribute each group to a classroom?

No!

Symmetry

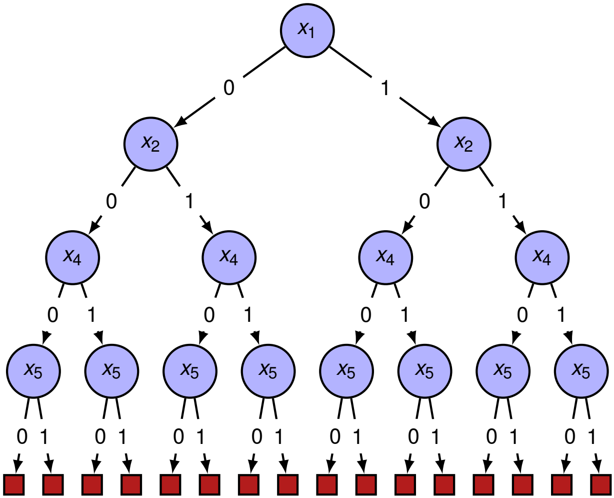
$$g = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\ x_2 & x_1 & x_3 & x_5 & x_4 & x_6 & x_8 & x_7 & x_9 \end{pmatrix} \rightarrow (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$$

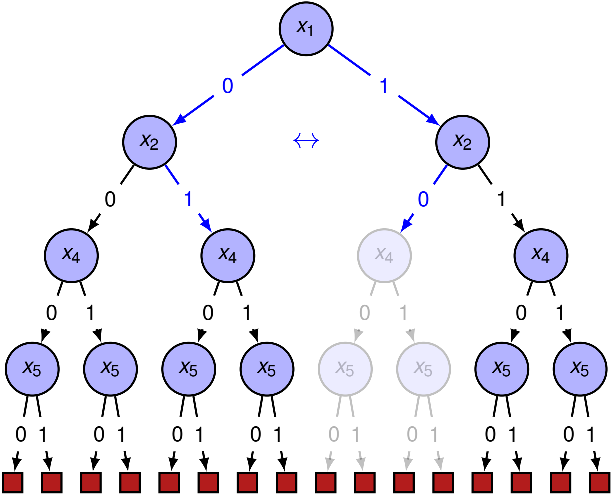
$$\begin{array}{lcl}
\omega_1 = \{x_1, x_2, x_3\} & & \omega_1 = \{x_2, x_1, x_3\} \\
\omega_2 = \{x_4, x_5, x_6\} & & \omega_2 = \{x_5, x_4, x_6\} \\
\omega_3 = \{x_7, x_8, x_9\} & & \omega_3 = \{x_8, x_7, x_9\} \\
\omega_4 = \{\neg x_1, \neg x_4\} & & \omega_7 = \{\neg x_2, \neg x_5\} \\
\omega_5 = \{\neg x_1, \neg x_7\} & & \omega_8 = \{\neg x_2, \neg x_8\} \\
\omega_6 = \{\neg x_4, \neg x_7\} & & \omega_9 = \{\neg x_5, \neg x_8\} \\
\omega_7 = \{\neg x_2, \neg x_5\} & & \omega_4 = \{\neg x_1, \neg x_4\} \\
\omega_8 = \{\neg x_2, \neg x_8\} & & \omega_5 = \{\neg x_1, \neg x_7\} \\
\omega_9 = \{\neg x_5, \neg x_8\} & & \omega_6 = \{\neg x_4, \neg x_7\} \\
\omega_{10} = \{\neg x_3, \neg x_6\} & & \omega_{10} = \{\neg x_3, \neg x_6\} \\
\omega_{11} = \{\neg x_3, \neg x_9\} & & \omega_{11} = \{\neg x_3, \neg x_9\} \\
\omega_{12} = \{\neg x_6, \neg x_9\} & & \omega_{12} = \{\neg x_6, \neg x_9\}
\end{array}$$

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

Outline

1. SAT overview
   - SAT basics
   - SAT and symmetries
2. Existing approaches
   - Static symmetry breaking
   - Dynamic symmetry breaking
3. Contribution and results


x_1
0
1
x_2
x_2
0
1
0
1
x_4
x_4
x_4
x_4
0 1 0 1 0 1 0 1
x_5 x_5 x_5 x_5 x_5 x_5 x_5 x_5
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1


x_1
0
1
$\leftrightarrow$
x_2
x_2
0
1
0
1
x_4
x_4
x_4
x_4
0 1
0 1
0 1
0 1
x_5
x_5
x_5
x_5
x_5
x_5
x_5
x_5
0 1
0 1
0 1
0 1
0 1
0 1
0 1
0 1

Static symmetry breaking

TODO
Tree example with hide symmetric search space
Easy to use, no modification of the solver
Inject additional constraints to the problem symmetry breaking predicates sbp
Different tools Shatter, BreakID

Example

TODO
Show search tree and remove symmetrical search space tree

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 !

Computing symmetries of a SAT problem

CNF formula

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9)$$
$$\wedge(\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7)$$
$$\wedge(\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8)$$
$$\wedge(\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9)$$

[1] http://www.tcs.hut.fi/Software/bliss/

[2] http://vlsicad.eecs.umich.edu/BK/SAUCY/

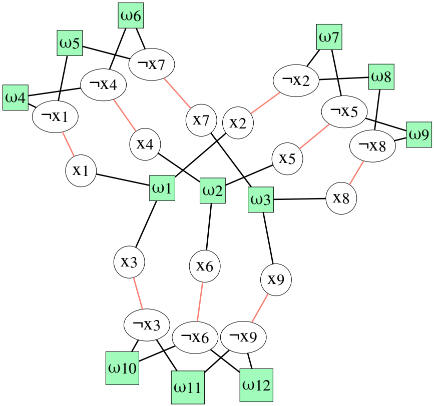
Computing symmetries of a SAT problem

CNF formula

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9)$$
$$\wedge(\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7)$$
$$\wedge(\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8)$$
$$\wedge(\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9)$$

$\Downarrow$ $\Downarrow$

colored graph



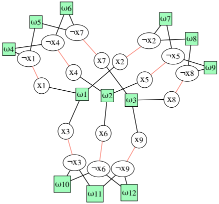
Computing symmetries of a SAT problem

CNF formula

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9)$$
$$\wedge(\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7)$$
$$\wedge(\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8)$$
$$\wedge(\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9)$$

$\Downarrow$ $\Downarrow$

colored graph



$\|$ $\|$

graph automorphism (`bliss` [1] or `saucy` [2])

[1] http://www.tcs.hut.fi/Software/bliss/

[2] http://vlsicad.eecs.umich.edu/BK/SAUCY/

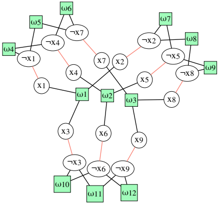
Computing symmetries of a SAT problem

CNF formula

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9)$$
$$\wedge(\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7)$$
$$\wedge(\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8)$$
$$\wedge(\neg x_3 \vee \neg x_6) \wedge (\neg x_3 \vee \neg x_9) \wedge (\neg x_6 \vee \neg x_9)$$

$\Downarrow$ $\Downarrow$

colored graph



$\Vert$ $\Vert$

graph automorphism (`bliss`[1] or `saucy`[2])

$\Downarrow$ $\Downarrow$

set of symmetries

$$g_1 = (x_2\ \ x_3)(x_5\ \ x_6)(x_8\ \ x_9)$$
$$g_2 = (x_4\ \ x_7)(x_5\ \ x_8)(x_6\ \ x_9)$$
$$g_3 = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$$
$$g_4 = (x_1\ \ x_4)(x_2\ \ x_5)(x_3\ \ x_6)$$

[1] http://www.tcs.hut.fi/Software/bliss/

[2] http://vlsicad.eecs.umich.edu/BK/SAUCY/

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.