Exploitation of dynamic symmetries for solving SAT problems

Doctorat de Sorbonne Université

Hakan METIN

Le, 18 décembre 2019

Rapporteurs:

PASCAL FONTAINE	Professeur, Université de Liège
LAURE PETRUCCI	Professeur, Université Paris 13

Examinateurs:

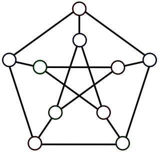
BART BOGAERTS	Assistant Professor, Vrije Universiteit Brussel
JEAN-MICHEL COUVREUR	Professeur, Université d'Orléans
EMMANUELLE ENCRENAZ	Maître de conférences, Sorbonne Université

Directeurs:

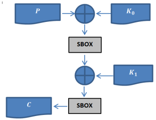
SOUHEIB BAARIR	Maître de conférences, Université Paris Nanterre
FABRICE KORDON	Professeur, Sorbonne Université

Motivation

Graph coloring

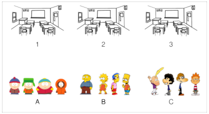


Cryptanalysis



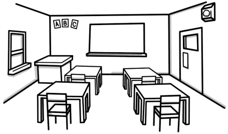
Hardware and sofware verification

Planning

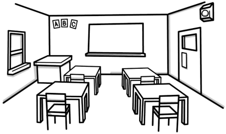


Boolean SATisfiability

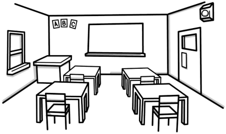
SAT: an example (1/2)



1

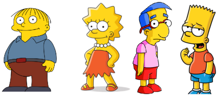


2

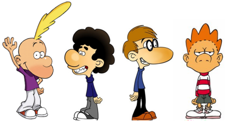


3

A



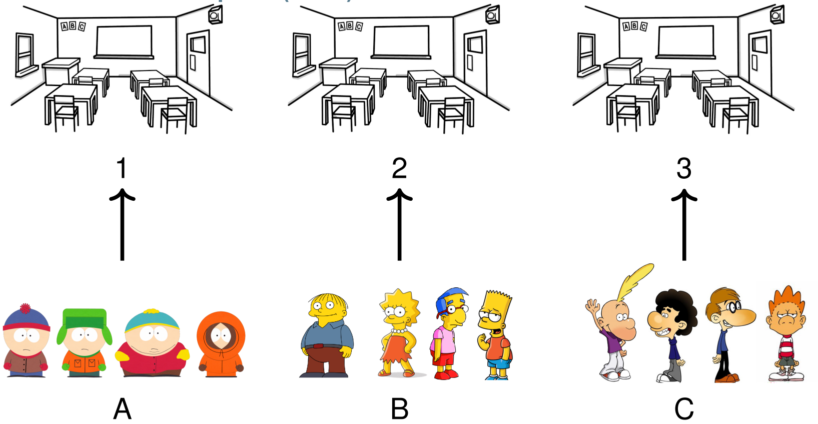
B



C

Is it possible to attribute each group to a unique classroom?

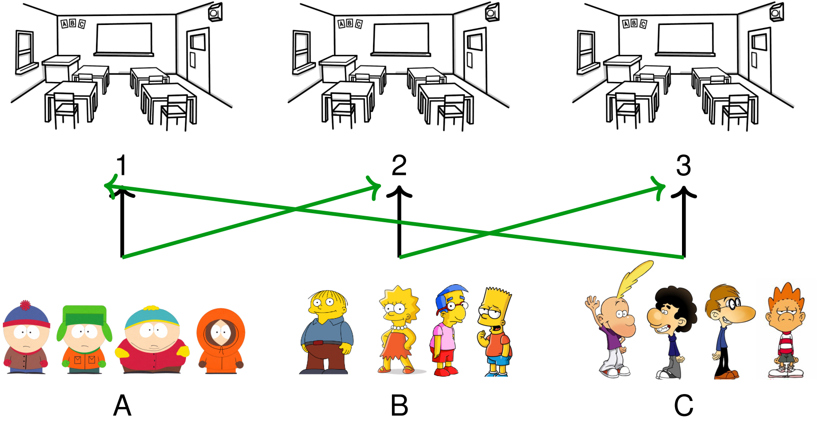
SAT: an example (1/2)



Is it possible to attribute each group to a unique classroom?

YES! $\texttt{SAT}$isfiable $\quad \alpha = \{(A, 1), (B, 2), (C, 3)\}$

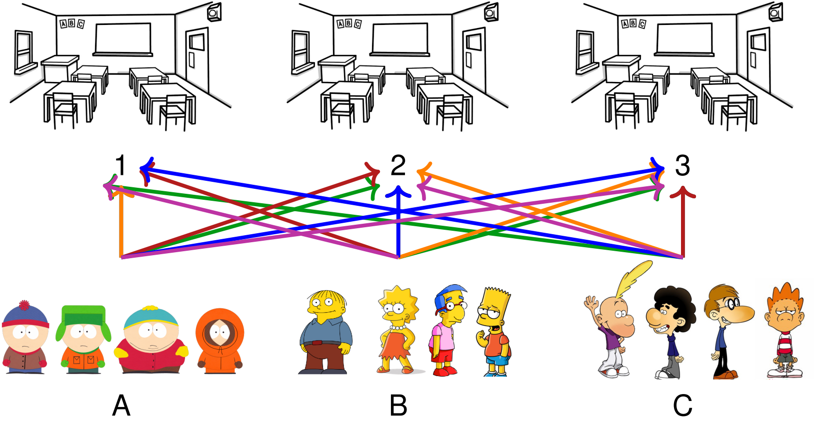
SAT: an example (1/2)



Is it possible to attribute each group to a unique classroom?

YES! $\texttt{SAT}$isfiable $\alpha = \{(A, 1), (B, 2), (C, 3)\}$
Many solutions $\alpha' = \{(A, 2), (B, 3), (C, 1)\}$

SAT: an example (1/2)

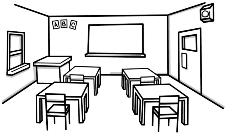


Is it possible to attribute each group to a unique classroom?

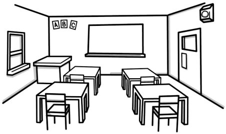
YES! SATisfiable $\alpha = \{(A, 1), (B, 2), (C, 3)\}$
Many solutions $\alpha' = \{(A, 2), (B, 3), (C, 1)\}$

$\vdots$

SAT: an example (2/2)

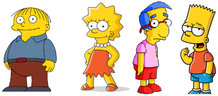


1

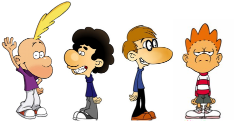


2

A



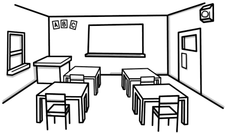
B



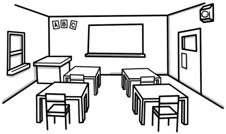
C

Is it possible to attribute each group to a unique classroom?

SAT: an example (2/2)

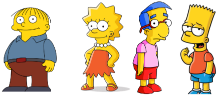


1

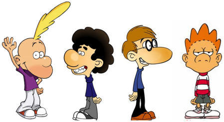


2

A



B

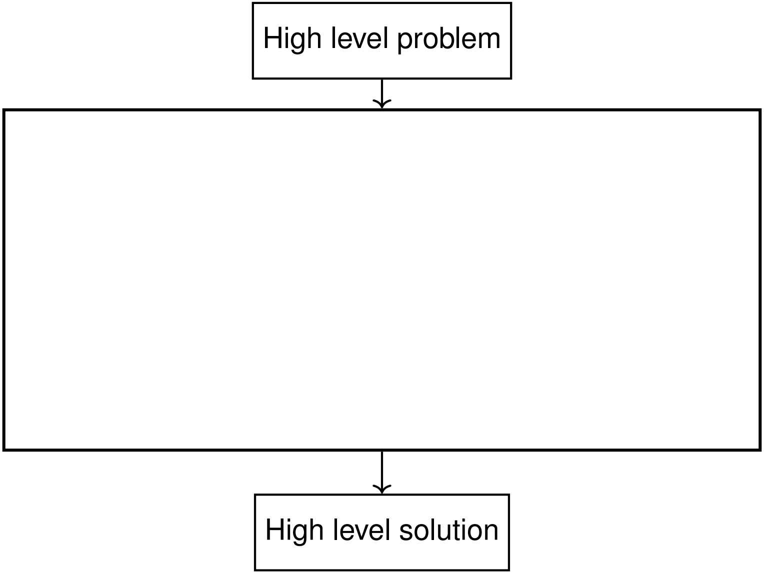


C

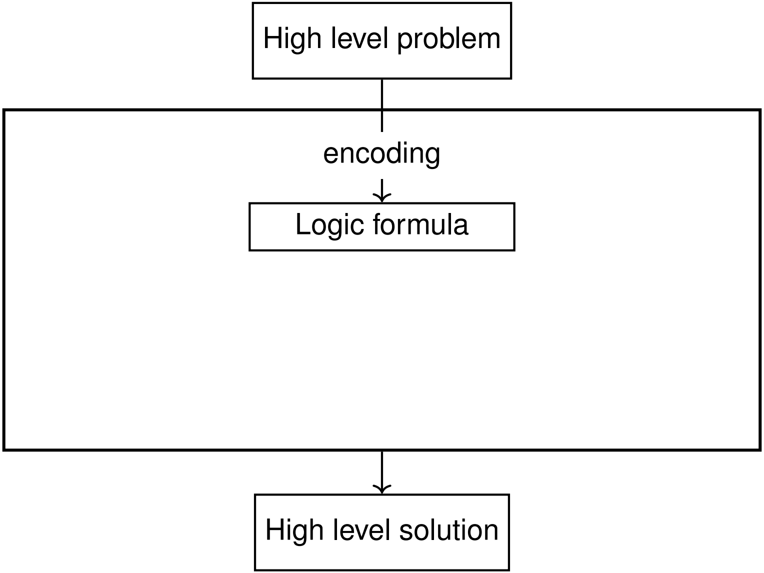
Is it possible to attribute each group to a unique classroom?

No! `UNSAT`isfiable

From high level problem to the solution through SAT solving



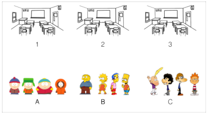
From high level problem to the solution through SAT solving



Encoding the problem

$$\overbrace{(A,1)}^{x_1}\overbrace{(A,2)}^{x_2}\overbrace{(A,3)}^{x_3}$$

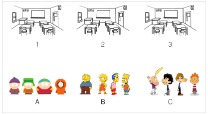
$$(x_1 \vee x_2 \vee x_3) \ \wedge$$



Encoding the problem

$$\begin{array}{ccc} \overbrace{(A,1)}^{x_1} & \overbrace{(A,2)}^{x_2} & \overbrace{(A,3)}^{x_3} \\ (B,1) & (B,2) & (B,3) \\ (C,1) & (C,2) & (C,3) \end{array}$$

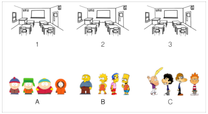
$$\begin{array}{l} (x_1 \vee x_2 \vee x_3) \ \wedge \\ (x_4 \vee x_5 \vee x_6) \ \wedge \\ (x_7 \vee x_8 \vee x_9) \ \wedge \end{array}$$



Encoding the problem

$$\overbrace{(A,1)}^{x_1}\overbrace{(A,2)}^{x_2}\overbrace{(A,3)}^{x_3}$$
$$(B,1)(B,2)(B,3)$$
$$(C,1)(C,2)(C,3)$$

$$\neg(A,1)\neg(B,1)$$
$$\neg(A,1)\neg(C,1)$$
$$\neg(B,1)\neg(C,1)$$



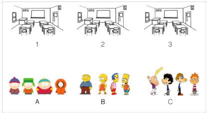
$$(x_1 \lor x_2 \lor x_3) \ \land$$
$$(x_4 \lor x_5 \lor x_6) \ \land$$
$$(x_7 \lor x_8 \lor x_9) \ \land$$

$$(\neg x_1 \lor \neg x_4) \ \land$$
$$(\neg x_1 \lor \neg x_7) \ \land$$
$$(\neg x_4 \lor \neg x_7) \ \land$$

Encoding the problem

$$\overbrace{(A,1)}^{x_1}\overbrace{(A,2)}^{x_2}\overbrace{(A,3)}^{x_3}$$
$$(B,1)(B,2)(B,3)$$
$$(C,1)(C,2)(C,3)$$

$$\neg(A,1)\neg(B,1)$$
$$\neg(A,1)\neg(C,1)$$
$$\neg(B,1)\neg(C,1)$$



$$\neg(A,2)\neg(B,2)$$
$$\neg(A,2)\neg(C,2)$$
$$\neg(B,2)\neg(C,2)$$

$$\neg(A,3)\neg(B,3)$$
$$\neg(A,3)\neg(C,3)$$
$$\neg(B,3)\neg(C,3)$$

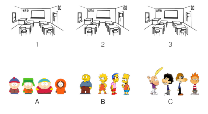
$$(x_1 \vee x_2 \vee x_3) \quad \wedge$$
$$(x_4 \vee x_5 \vee x_6) \quad \wedge$$
$$(x_7 \vee x_8 \vee x_9) \quad \wedge$$

$$(\neg x_1 \vee \neg x_4) \quad \wedge$$
$$(\neg x_1 \vee \neg x_7) \quad \wedge$$
$$(\neg x_4 \vee \neg x_7) \quad \wedge$$

$$(\neg x_2 \vee \neg x_5) \quad \wedge$$
$$(\neg x_2 \vee \neg x_8) \quad \wedge$$
$$(\neg x_5 \vee \neg x_8) \quad \wedge$$

$$(\neg x_3 \vee \neg x_6) \quad \wedge$$
$$(\neg x_3 \vee \neg x_9) \quad \wedge$$
$$(\neg x_6 \vee \neg x_9)$$

Encoding the problem



$$\overbrace{(A,1)}^{x_1}\overbrace{(A,2)}^{x_2}\overbrace{(A,3)}^{x_3}$$
$$(B,1)(B,2)(B,3)$$
$$(C,1)(C,2)(C,3)$$

$$\neg(A,1)\neg(B,1)$$
$$\neg(A,1)\neg(C,1)$$
$$\neg(B,1)\neg(C,1)$$

$$\neg(A,2)\neg(B,2)$$
$$\neg(A,2)\neg(C,2)$$
$$\neg(B,2)\neg(C,2)$$

$$\neg(A,3)\neg(B,3)$$
$$\neg(A,3)\neg(C,3)$$
$$\neg(B,3)\neg(C,3)$$

Clause

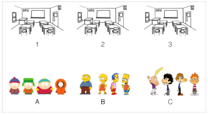
$$\boxed{(x_1 \lor x_2 \lor x_3)} \land$$
$$(x_4 \lor x_5 \lor x_6) \land$$
$$(x_7 \lor x_8 \lor x_9) \land$$

$$(\neg x_1 \lor \neg x_4) \land$$
$$(\neg x_1 \lor \neg x_7) \land$$
$$(\neg x_4 \lor \neg x_7) \land$$

$$(\neg x_2 \lor \neg x_5) \land$$
$$(\neg x_2 \lor \neg x_8) \land$$
$$(\neg x_5 \lor \neg x_8) \land$$

$$(\neg x_3 \lor \neg x_6) \land$$
$$(\neg x_3 \lor \neg x_9) \land$$
$$(\neg x_6 \lor \neg x_9)$$

Encoding the problem



$$\underbrace{(A,1)}^{x_1}\underbrace{(A,2)}^{x_2}\underbrace{(A,3)}^{x_3}$$
$$(B,1)(B,2)(B,3)$$
$$(C,1)(C,2)(C,3)$$

$$\neg(A,1)\neg(B,1)$$
$$\neg(A,1)\neg(C,1)$$
$$\neg(B,1)\neg(C,1)$$

$$\neg(A,2)\neg(B,2)$$
$$\neg(A,2)\neg(C,2)$$
$$\neg(B,2)\neg(C,2)$$

$$\neg(A,3)\neg(B,3)$$
$$\neg(A,3)\neg(C,3)$$
$$\neg(B,3)\neg(C,3)$$

Clause

$$\boxed{(x_1 \vee x_2 \vee x_3)} \wedge$$
$$(x_4 \vee x_5 \vee x_6) \wedge$$
$$(x_7 \vee x_8 \vee x_9) \wedge$$

$$(\neg x_1 \vee \neg x_4) \wedge$$
$$(\neg x_1 \vee \neg x_7) \wedge$$
$$(\neg x_4 \vee \neg x_7) \wedge$$

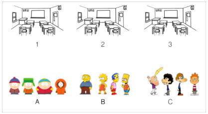
$$(\neg x_2 \vee \neg x_5) \wedge$$
$$(\neg x_2 \vee \neg x_8) \wedge$$
$$(\neg x_5 \vee \neg x_8) \wedge$$

$$(\neg x_3 \vee \neg x_6) \wedge$$
$$(\neg x_3 \vee \neg x_9) \wedge$$
$$(\neg x_6 \vee \neg x_9)$$

$\leftarrow$ Conjunctive Normal Form (CNF)

Any Boolean formula can be transformed into CNF in polynomial time

Encoding the problem



$$\overbrace{(A,1)}^{x_1}\ \overbrace{(A,2)}^{x_2}\ \overbrace{(A,3)}^{x_3}$$
$$(B,1)(B,2)(B,3)$$
$$(C,1)(C,2)(C,3)$$

$$\neg(A,1)\neg(B,1)$$
$$\neg(A,1)\neg(C,1)$$
$$\neg(B,1)\neg(C,1)$$

$$\neg(A,2)\neg(B,2)$$
$$\neg(A,2)\neg(C,2)$$
$$\neg(B,2)\neg(C,2)$$

$$\neg(A,3)\neg(B,3)$$
$$\neg(A,3)\neg(C,3)$$
$$\neg(B,3)\neg(C,3)$$

Clause

$$\boxed{(x_1 \vee x_2 \vee x_3)} \wedge$$
$$(x_4 \vee x_5 \vee x_6) \wedge$$
$$(x_7 \vee x_8 \vee x_9) \wedge$$

$$(\neg x_1 \vee \neg x_4) \wedge$$
$$(\neg x_1 \vee \neg x_7) \wedge$$
$$(\neg x_4 \vee \neg x_7) \wedge$$

$$(\neg x_2 \vee \neg x_5) \wedge$$
$$(\neg x_2 \vee \neg x_8) \wedge$$
$$(\neg x_5 \vee \neg x_8) \wedge$$

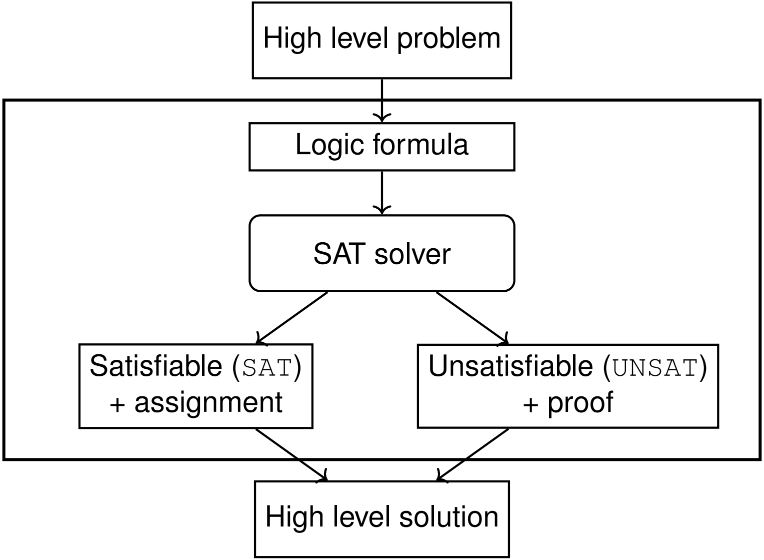
$$(\neg x_3 \vee \neg x_6) \wedge$$
$$(\neg x_3 \vee \neg x_9) \wedge$$
$$(\neg x_6 \vee \neg x_9)$$

Conjunctive Normal Form (CNF)

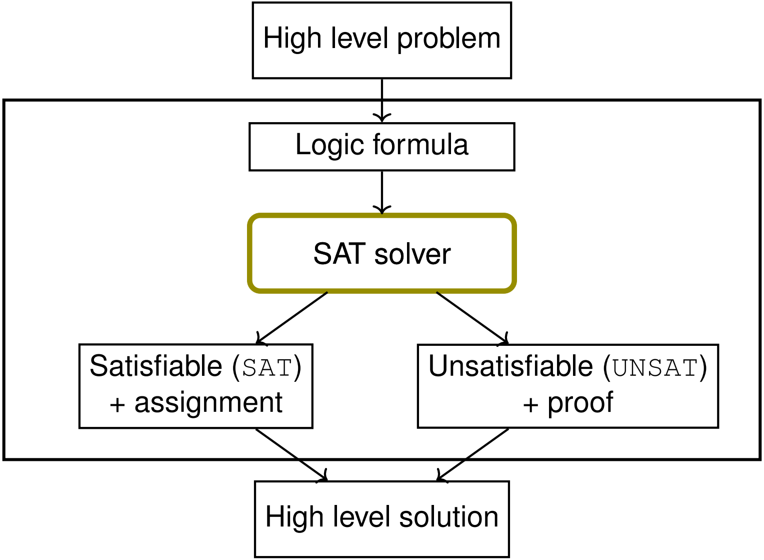
Clause represented as a set:

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

From high level problem to the solution through SAT solving



From high level problem to the solution through SAT solving



SAT Solving

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

Good performance in practice:

- Handle large problem (million variables and clauses)
- International SAT competition each year on academic and industrial problems

SAT Solving

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

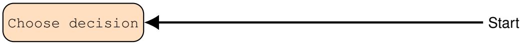
Good performance in practice:

- Handle large problem (million variables and clauses)
- International SAT competition each year on academic and industrial problems

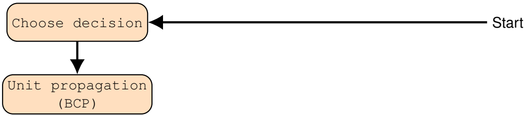
Enumerative algorithms:

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

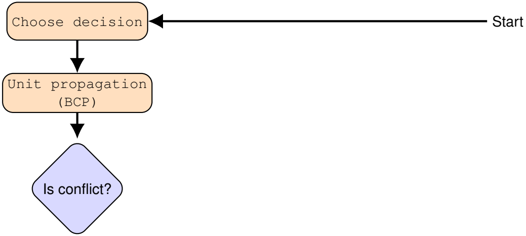
CDCL in detail



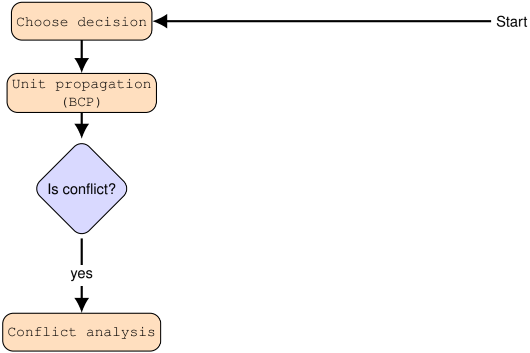
CDCL in detail



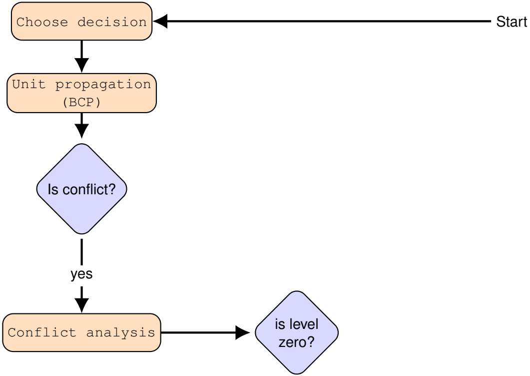
CDCL in detail



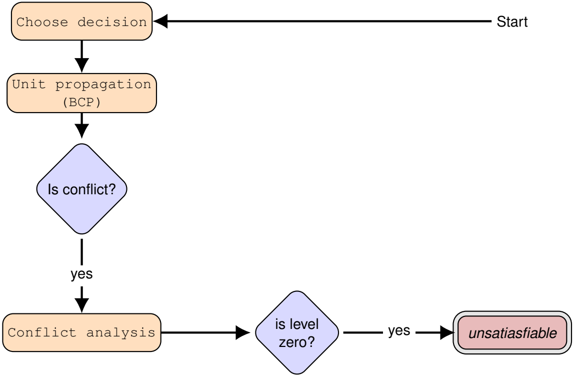
CDCL in detail



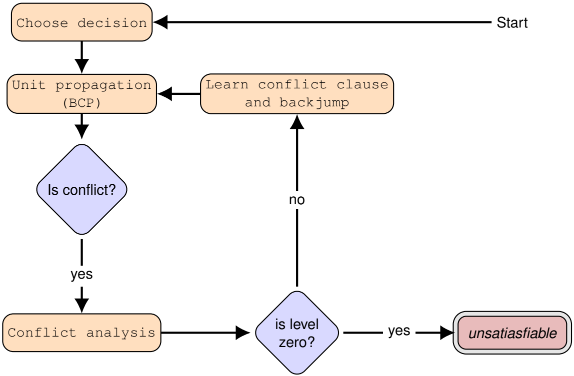
CDCL in detail



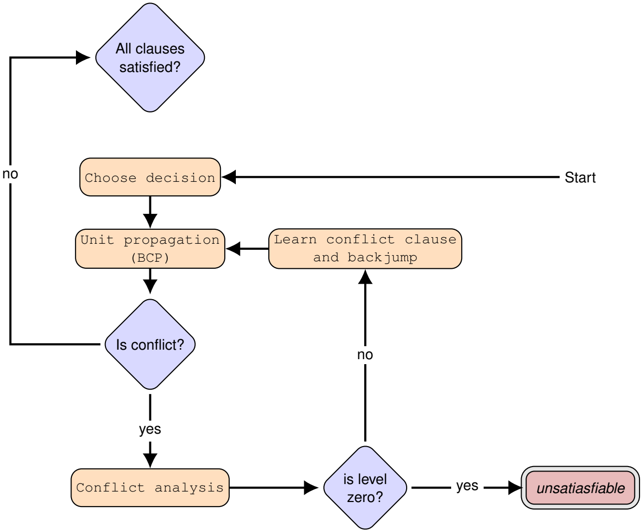
CDCL in detail



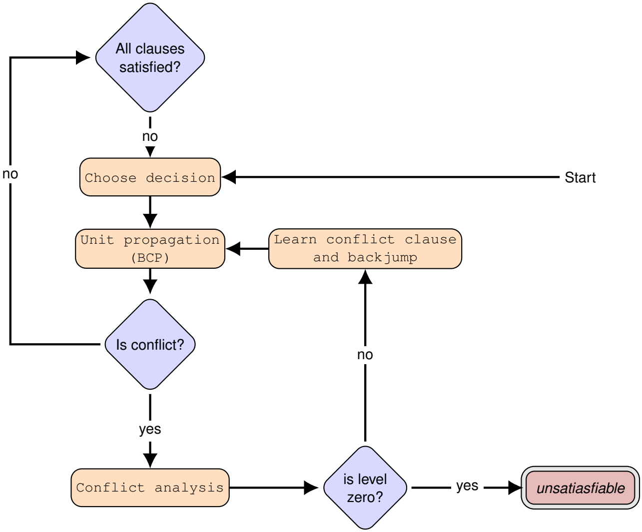
CDCL in detail



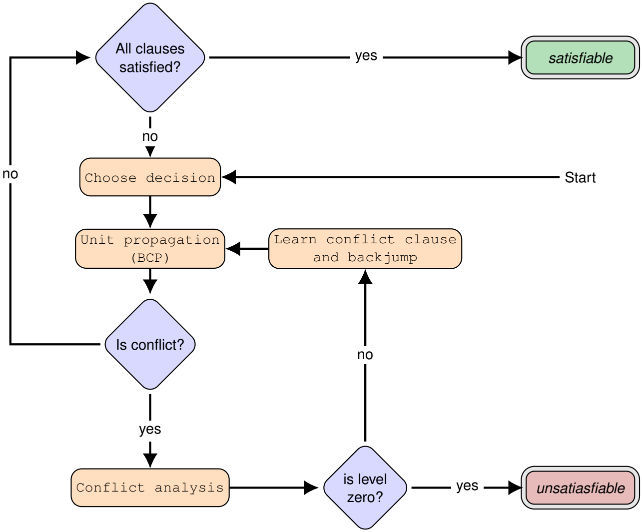
CDCL in detail



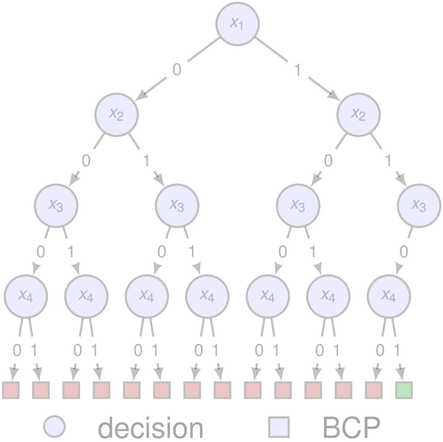
CDCL in detail



CDCL in detail



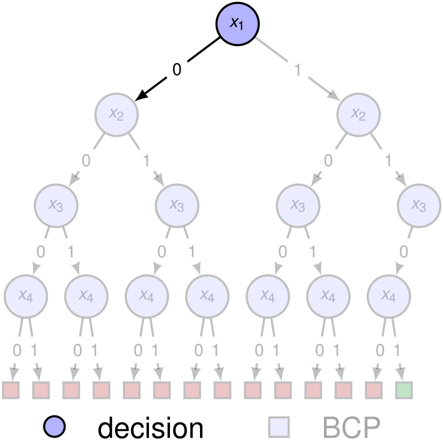
CDCL in action



$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$

$$\alpha = \{\}$$

CDCL in action

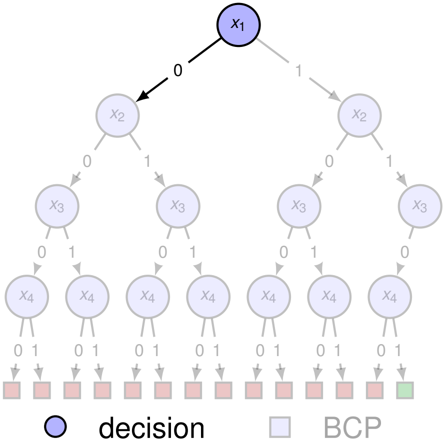


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$

$$\alpha = \{\neg x_1\}$$

CDCL in action

Conflict Analysis

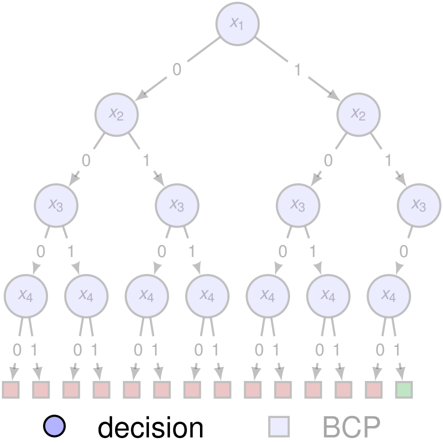


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$

$$\alpha = \{\neg x_1\}$$

CDCL in action

Learn conflict clause and backjump

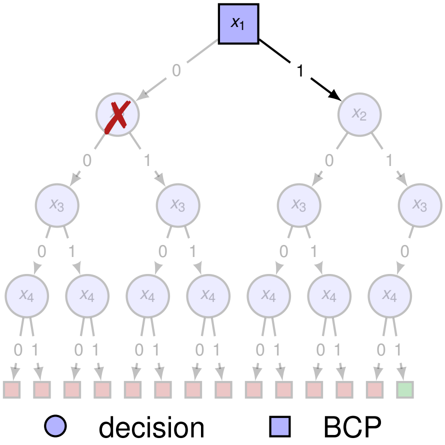


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$

$$\alpha = \{\}$$

CDCL in action

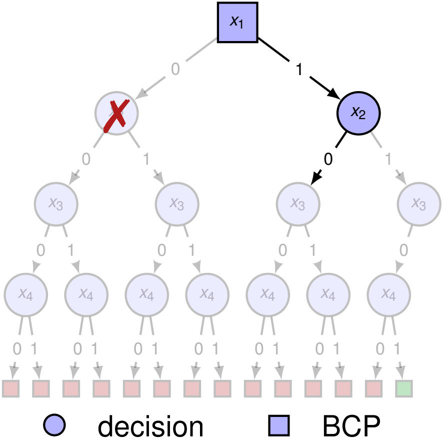
Unit Propagation (BCP)



$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$

$$\alpha = \{x_1\}$$

CDCL in action

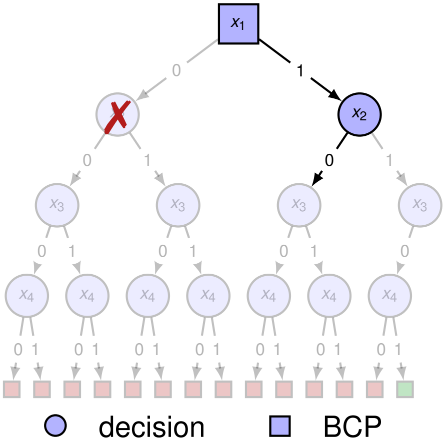


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$

$$\alpha = \{x_1, \neg x_2\}$$

CDCL in action

Conflict Analysis

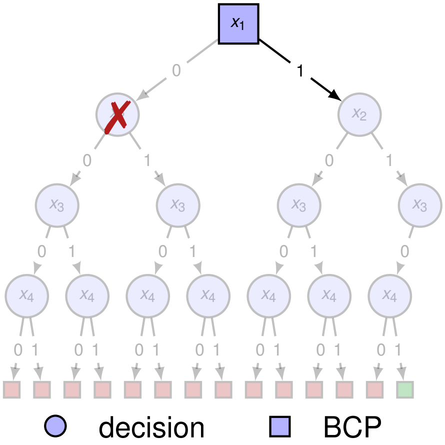


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$

$$\alpha = \{x_1, \neg x_2\}$$

CDCL in action

Learn conflict clause and backjump

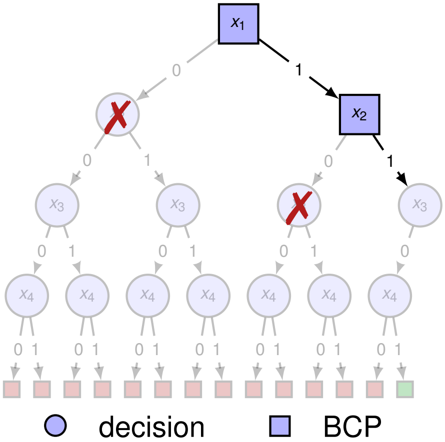


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$
$$\omega_8 = \{x_2\}$$

$$\alpha = \{x_1\}$$

CDCL in action

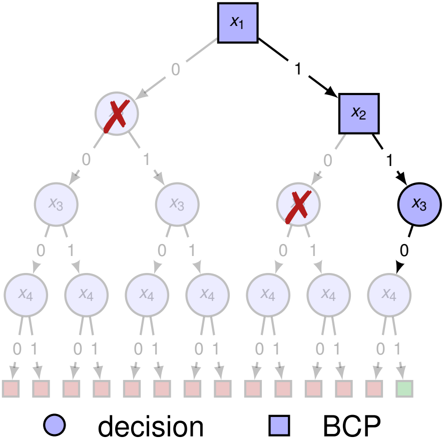
Unit Propagation (BCP)



$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$
$$\omega_8 = \{x_2\}$$

$$\alpha = \{x_1, x_2\}$$

CDCL in action

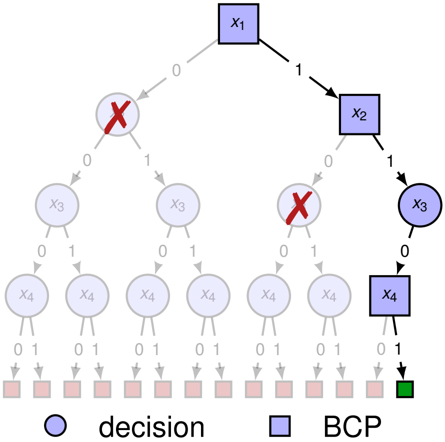


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$
$$\omega_8 = \{x_2\}$$

$$\alpha = \{x_1, x_2, \neg x_3\}$$

CDCL in action

satisfiable

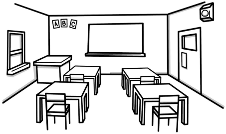


$$\omega_1 = \{x_1, x_2, x_3, x_4\}$$
$$\omega_2 = \{x_1, \neg x_4\}$$
$$\omega_3 = \{x_1, x_4\}$$
$$\omega_4 = \{x_2, \neg x_4\}$$
$$\omega_5 = \{x_2, x_4\}$$
$$\omega_6 = \{x_3, x_4\}$$
$$\omega_7 = \{x_1\}$$
$$\omega_8 = \{x_2\}$$

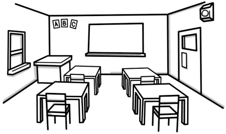
$$\alpha = \{x_1, x_2, \neg x_3, x_4\}$$

SAT and symmetries

Presence of symmetries

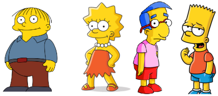


1

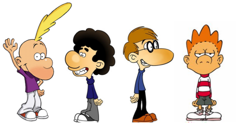


2

A

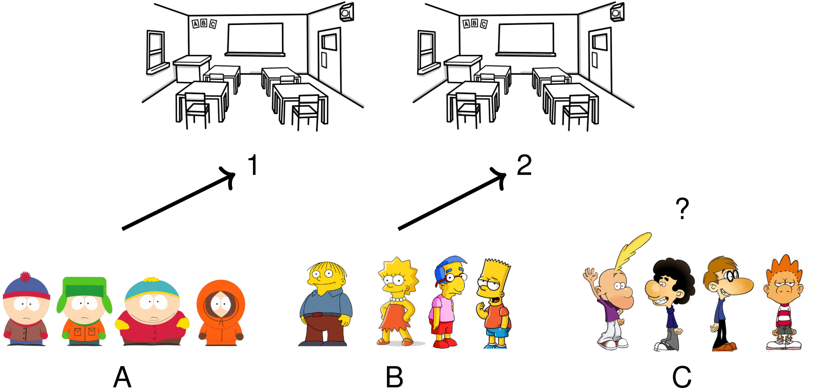


B

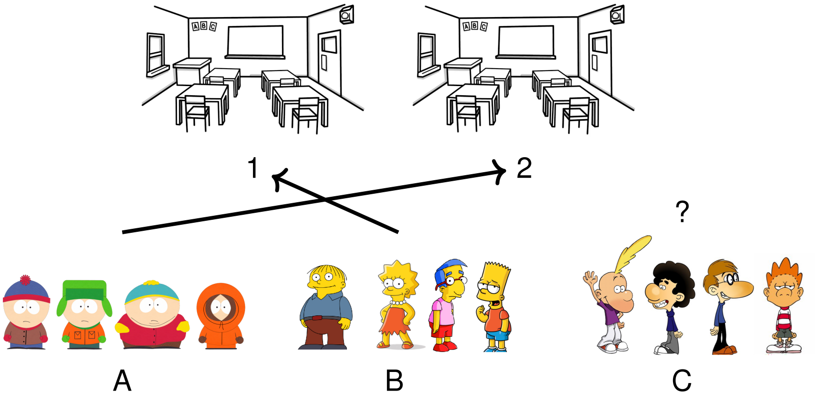


C

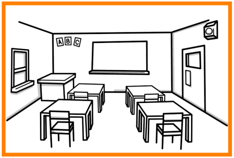
Presence of symmetries



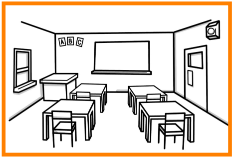
Presence of symmetries



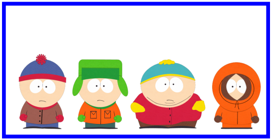
Presence of symmetries



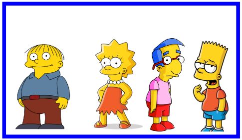
1



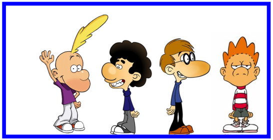
2



A

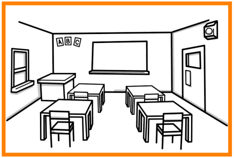


B

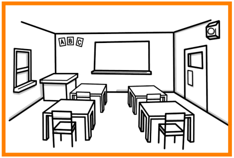


C

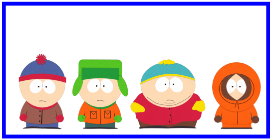
Presence of symmetries



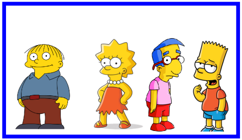
~~1~~ 2



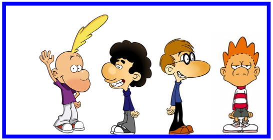
~~2~~ 1



A



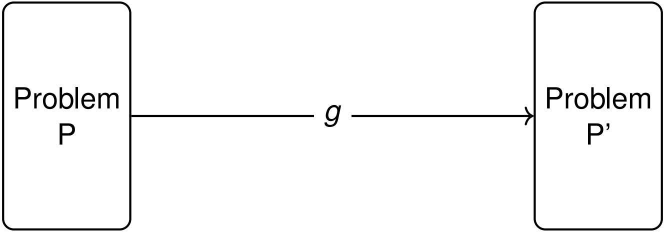
~~B~~ C



~~C~~ B

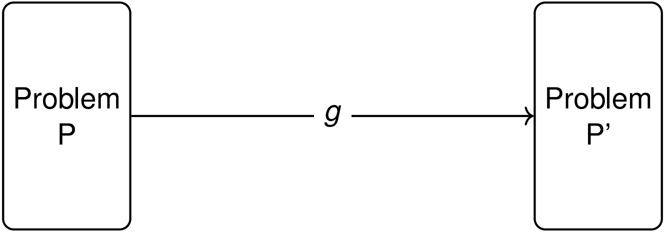
Symmetry in high level

g: a symmetry



Symmetry in high level

g: a symmetry

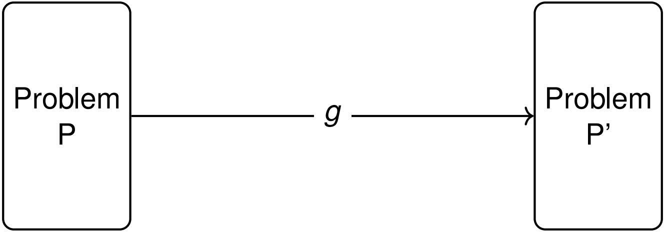


Equi-satisfiability

$$solution \models P \Leftrightarrow g.solution \models P'$$

Symmetry in high level

g: a symmetry



Equi-satisfiability

$$solution \models P \Leftrightarrow g.solution \models P'$$

- Semantic symmetries
- **Syntactic symmetries**

Syntactic symmetry

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

Syntactic symmetry

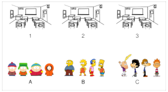
A symmetry (permuation) g is a bijective function (on variables) that leaves the formula φ invariant

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

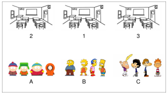
Syntactic symmetry

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

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



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



P

g.P = P' = P

Computing symmetries of a SAT problem

CNF formula

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

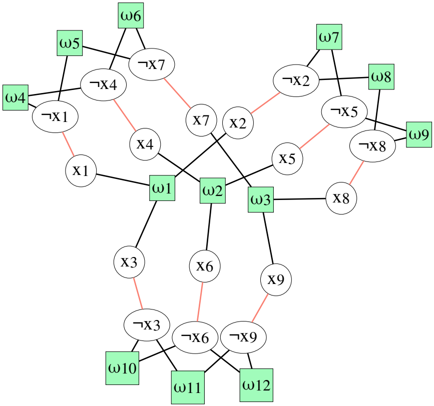
Computing symmetries of a SAT problem

CNF formula

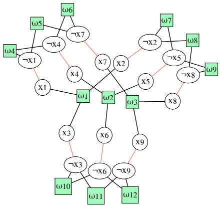
$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9)$$
$$\wedge(\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_7) \wedge (\neg x_4 \vee \neg x_7)$$
$$\wedge(\neg x_2 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_8) \wedge (\neg x_5 \vee \neg x_8)$$
$$\wedge(\neg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

$\Downarrow$ $\Downarrow$

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	
$\parallel$	$\parallel$
graph automorphism	(`bliss`, `saucy`, ...)

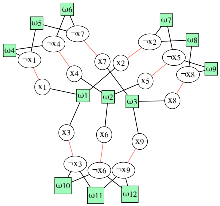
Computing symmetries of a SAT problem

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



$\|$ $\|$

graph automorphism (`bliss`, `saucy`, ...)

$\Downarrow$ $\Downarrow$

set of symmetries

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

The set of symmetries of a formula is a group noted $< G, \circ >$

Exploitation of symmetries:

Static symmetry breaking

Orbit

Orbit of an assignment α for a group G:

$$G.\alpha = \{g.\alpha \mid g \in G\}$$

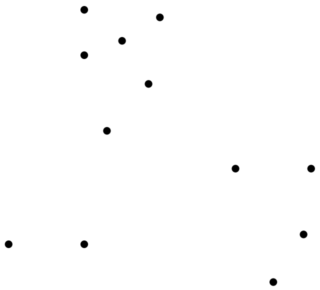
Orbit

Orbit of an assignment α for a group G:

$$G.\alpha = \{g.\alpha \mid g \in G\}$$

Example:

- full assignment



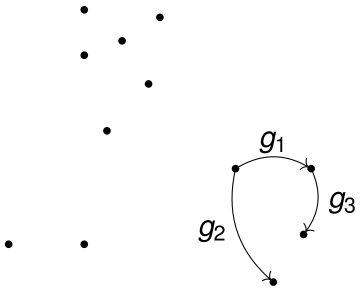
Orbit

Orbit of an assignment α for a group G:

$$G.\alpha = \{g.\alpha \mid g \in G\}$$

Example:

- full assignment



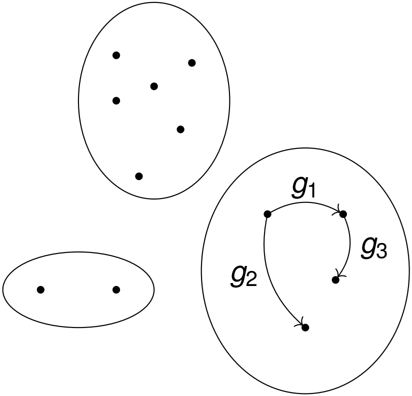
Orbit

Orbit of an assignment α for a group G:

$$G.\alpha = \{g.\alpha \mid g \in G\}$$

Example:

- full assignment
- orbit



Equivalence relation with respect to SAT:

- Either $G.\alpha$ contains no solution
- Or all elements of $G.\alpha$ are solutions

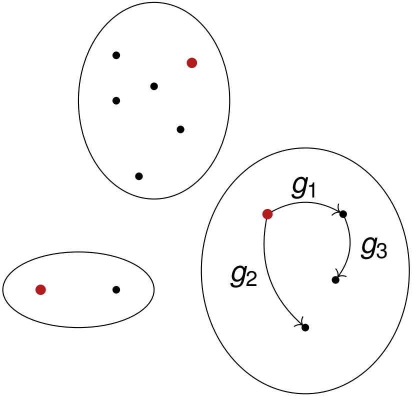
Orbit

Orbit of an assignment α for a group G:

$$G.\alpha = \{g.\alpha \mid g \in G\}$$

Example:

- full assignment
- orbit
- representative



Equivalence relation with respect to SAT:

- Either $G.\alpha$ contains no solution
- Or all elements of $G.\alpha$ are solutions

Comparing assignments: Assessments

Define an ordering relation to compare assignments ($\prec$)

- Total ordering on variables
- Minimum value: $\texttt{F} < \texttt{T}$ or $\texttt{T} < \texttt{F}$

Comparing assignments: Assessments

Define an ordering relation to compare assignments ($\prec$)

- Total ordering on variables
- Minimum value: $\texttt{F} < \texttt{T}$ or $\texttt{T} < \texttt{F}$

Allow only minimal value (lex-leader)

Comparing assignments: Assessments

Define an ordering relation to compare assignments ($\prec$)

- Total ordering on variables
- Minimum value: $\mathtt{F} < \mathtt{T}$ or $\mathtt{T} < \mathtt{F}$

Allow only minimal value (lex-leader)

Forbid other assignments in each orbit
$\rightarrow$ Add all symmetry breaking predicates (SBP) statically

Comparing assignments: Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$

Assignments:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	F	F	F	F	F	F

Comparing assignments: Example

Ordering relation: $x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le x_6 \le x_7 \le x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$

Assignments:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	F	F	F	F	F	F
$g.\alpha$	F	T	F	F	F	F	F	F

Comparing assignments: Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$

Assignments:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	F	F	F	F	F	F
$g.\alpha$	F	T	F	F	F	F	F	F

Comparing:

$$g.\alpha \prec \alpha$$

Comparing assignments: Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$

Assignments:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	F	F	F	F	F	F
$g.\alpha$	F	T	F	F	F	F	F	F

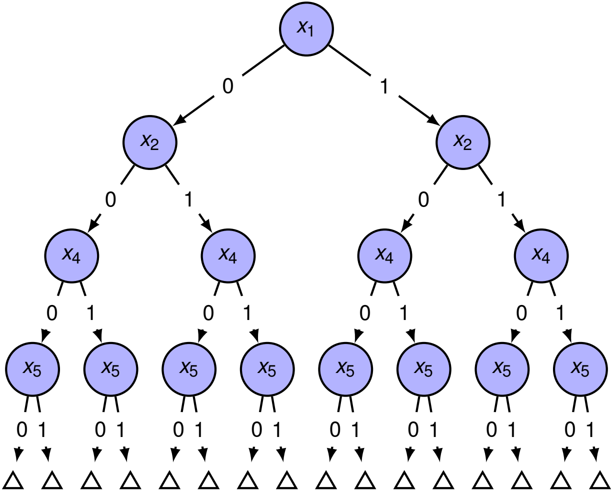
Comparing:

$$g.\alpha \prec \alpha \Rightarrow \text{SBP: } \omega = \{\neg x_1, x_2\}$$

Using symmetries to prune the search space

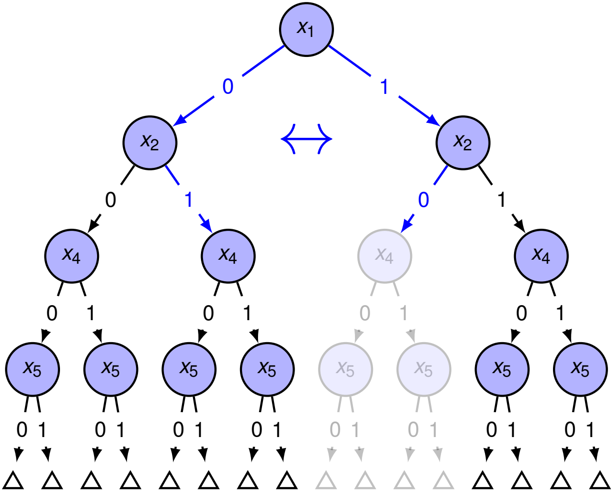
$g = (x_1 \ \ x_2)(x_4 \ \ x_5)(x_7 \ \ x_8)$

$\omega = \{\neg x_1, x_2\}$



Using symmetries to prune the search space

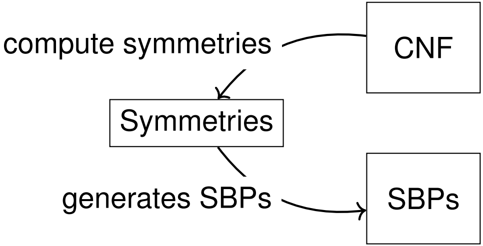
$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$ $\qquad\qquad$ $\omega = \{\neg x_1, x_2\}$



State-of-the-art static symmetry breaking

State-of-the-art:

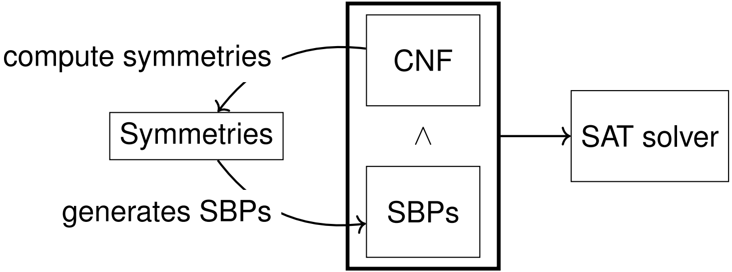
- Shatter [ASM06]
- BreakID [DBBD16]



State-of-the-art static symmetry breaking

State-of-the-art:

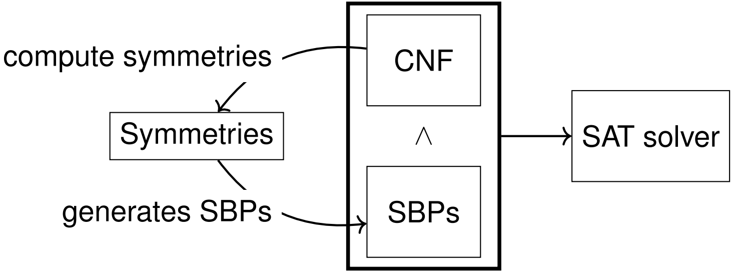
- Shatter [ASM06]
- BreakID [DBBD16]



State-of-the-art static symmetry breaking

State-of-the-art:

- Shatter [ASM06]
- BreakID [DBBD16]

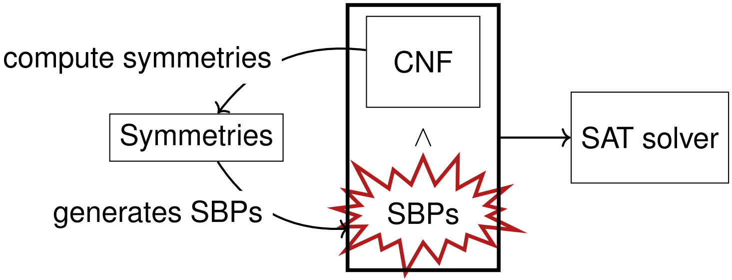


Works well on many symmetrical problems

State-of-the-art static symmetry breaking

State-of-the-art:

- Shatter [ASM06]
- BreakID [DBBD16]



Works well on many symmetrical problems

The solver can ”explode” instead of being helped

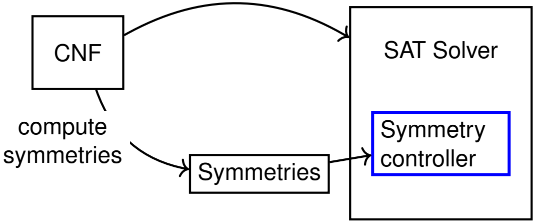
- generate not needed clause
- flooding the solver

First contribution:

CDCL[sym] Introducing Effective Symmetry Breaking in SAT Solving

TACAS'18 [MBCK18]

General idea

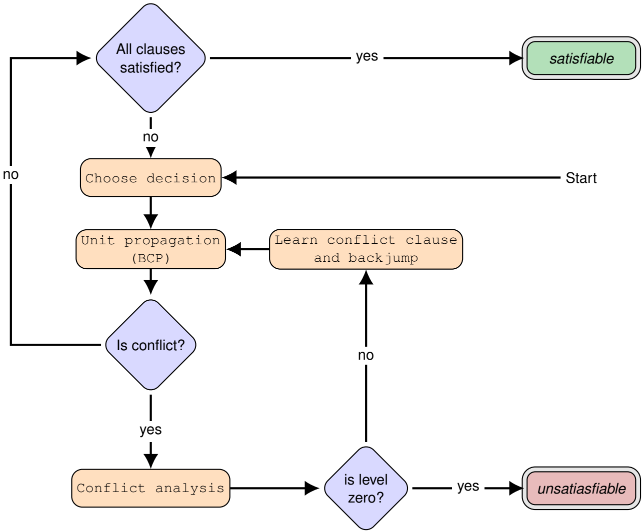


Symmetry controller:

- Generates SBP on-the-fly
- Only when needed
- Intrusive on solver

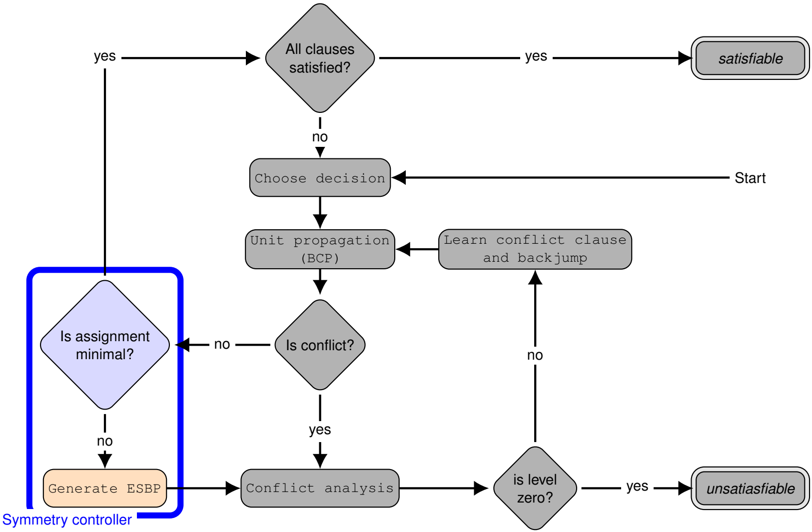
CDCL[Sym]

Compute and inject SBP opportunistically, during the solving



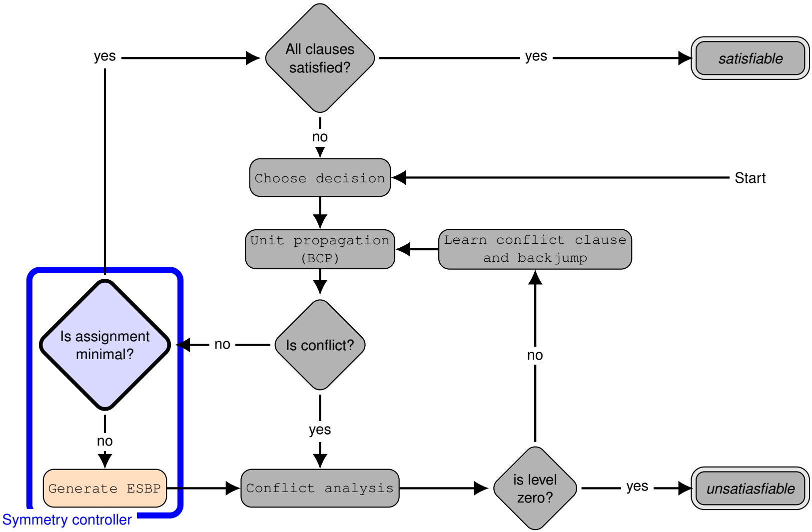
CDCL[Sym]

Compute and inject SBP opportunistically, during the solving



CDCL[Sym]

Compute and inject SBP opportunistically, during the solving



Is assignment minimal?

Our proposal: Symmetry status tracking

- `reducer`: $g.\alpha \prec \alpha$
- `inactive`: $\alpha \prec g.\alpha$
- `active`: *not enough information*

Is assignment minimal?

Our proposal: Symmetry status tracking

- `reducer`: $g.\alpha \prec \alpha$
- `inactive`: $\alpha \prec g.\alpha$
- `active`: *not enough information*

Efficient implementation of symmetry status tracking

Keep track the smallest unassigned variable x:

1. $\alpha(g.x) \leq \alpha(x)$, then g is `reducer` $\Rightarrow$ Effective SBP (ESBP)
2. $\alpha(x) \leq \alpha(g.x)$, then g is `inactive` $\Rightarrow$ g cannot reduce α
3. $\alpha(g.x)$ or $\alpha(x)$ is unassigned then g is `active`

Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

	$\downarrow$ x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	U	U	U	U	U	U	U	U
$g.\alpha$	U	U	U	U	U	U	U	U

$g.\alpha$ $\qquad \alpha$

status of permutation g: `active`

Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

	$\downarrow$ x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	U	F	U	U	U	U	U	U
$g.\alpha$	F	U	U	U	U	U	U	U

$g.\alpha \qquad \alpha$

status of permutation g: `active`

Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

	$\downarrow$ x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	U	F	U	F	U	U	U	U
$g.\alpha$	F	U	U	U	F	U	U	U

$g.\alpha \qquad \alpha$

status of permutation g: `active`

Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8$; $\mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

	$\downarrow$ x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	U	F	U	U	U	U
$g.\alpha$	F	T	U	U	F	U	U	U

$$g.\alpha \prec \alpha$$

status of permutation g: `reducer`

Example

Ordering relation: $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$

Symmetry: $g = (x_1 \ \ x_2)(x_4 \ \ x_5)(x_7 \ \ x_8)$

$$\downarrow$$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
α	T	F	U	F	U	U	U	U
$g.\alpha$	F	T	U	U	F	U	U	U

$$g.\alpha \prec \alpha$$

status of permutation g: `reducer`

On-the-fly generation of ESBP: $\omega = \{\neg x_1, x_2\}$

CDCL[Sym] implementation

- C++ Implementation: 1780 Loc
- Packaged as a library **cosy** (Controller of Symmetry)
  `https://github.com/lip6/cosy`
- Low memory consumption

- Virtually works with any enumerative CDCL SAT solver
- Can be easily integrated

$\rightarrow$ e.g. +3% LOC on `MiniSAT`
90 lines out of 3090

Experiments

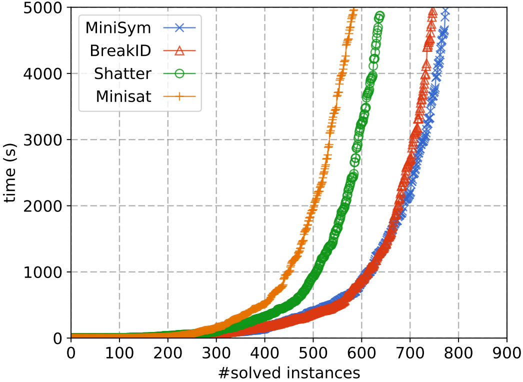
Benchmark:

- from SAT contests 2012 – 2017
- filter: `bliss` finds symmetries in 1000 seconds
- 36 % of instances, $1\,350/3\,700$

Setup:

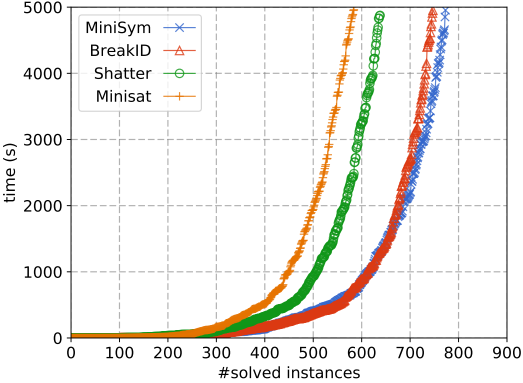
- four tools
  - MiniSat (no symmetry, baseline)
  - MiniSat + BreakID (SOTA SAT solver using symmetries)
  - MiniSat + Shatter (SOTA SAT solver using symmetries)
  - **MiniSym** = MiniSat + cosy (our approach)
- 5000 seconds timeout, 8GB memory
- includes time to compute symmetries (except for MiniSat)

Experimental results



Solver	PAR-2	SAT	UNSAT
MiniSAT	2243h	325	261
Shatter	2088h	316	324
BreakID	1790h	334	415
MiniSym	**1735h**	**336**	**439**

Experimental results



Number of SBPs	BreakID	MiniSym
UNSAT (399)	2 576 349	**913 339**
SAT (320)	12 179 513	**457 452**

Discussion of the results

Change the ordering relation

- Choose another lex-leader
- Generate other SBP

Discussion of the results

Change the ordering relation

- Choose another lex-leader
- Generate other SBP

Composing permutations

- Observe more variables
- Earlier generation of ESBP

Discussion of the results

Change the ordering relation

- Choose another lex-leader
- Generate other SBP

Composing permutations

- Observe more variables
- Earlier generation of ESBP

Adapt the solver heuristics dynamically

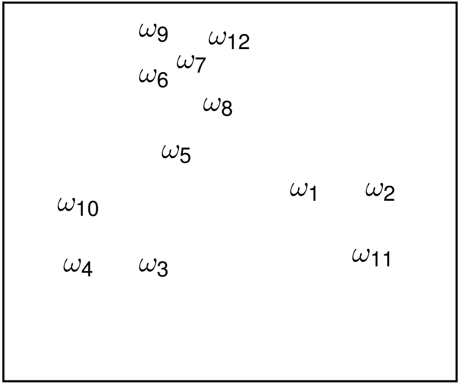
- Restart
- Cleaning database

Exploitation of symmetries:

Dynamic symmetry breaking

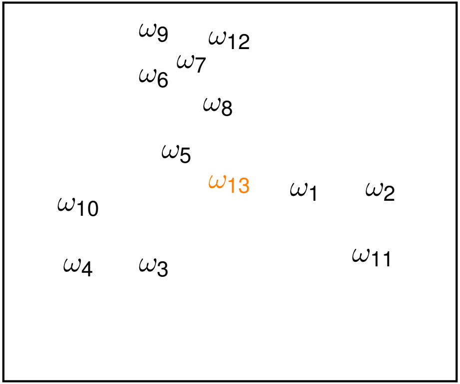
Learn symmetrical clauses

- □ formula
- ω clause



Learn symmetrical clauses

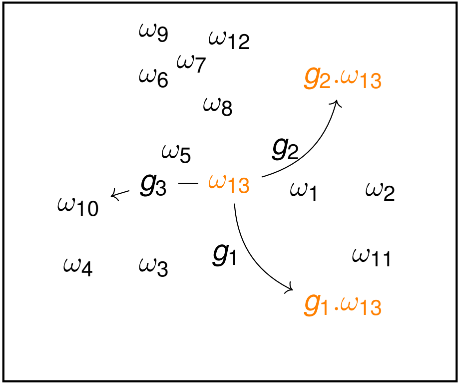
$\square$	formula
ω	clause
ω	learnt clause



Learnt clauses are logical consequences of the formula

Learn symmetrical clauses

- ▫ formula
- ω clause
- ω learnt clause

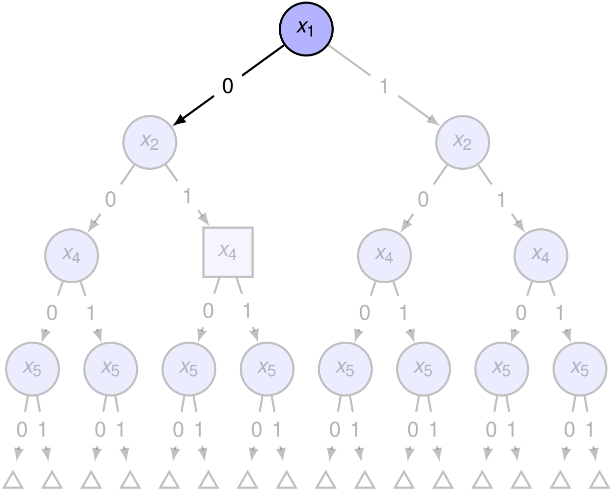


Learnt clauses are logical consequences of the formula

$g \in G$ are symmetries of the formula
$\rightarrow$ symmetrical learnt clauses are logical consequences too

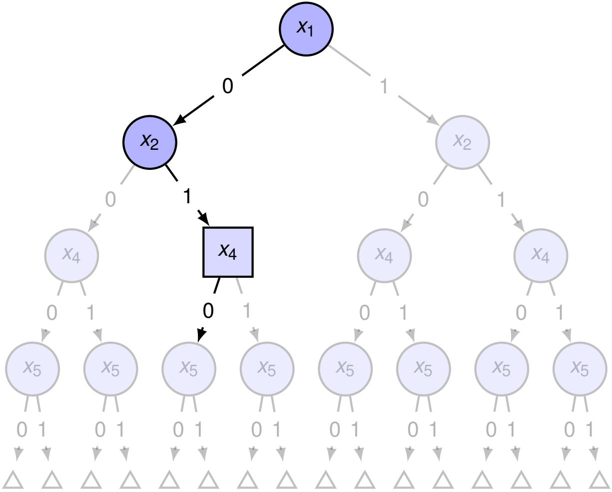
Using symmetries to accelerate the tree traversal

$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$



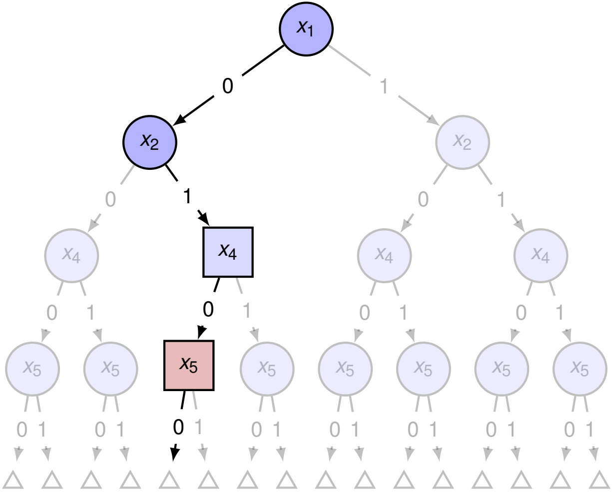
Using symmetries to accelerate the tree traversal

$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$



Using symmetries to accelerate the tree traversal

$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

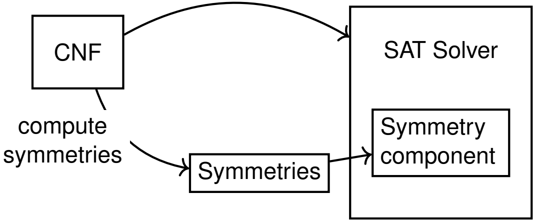


Use symmetries to deduce symmetrical facts.

State-of-the-art dynamic symmetry breaking

State-of-the-art:

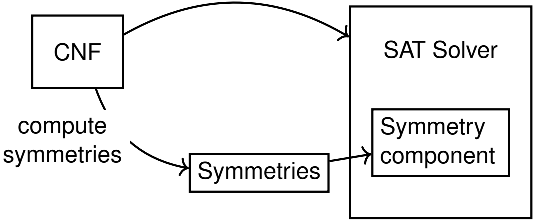
- Symmchaff [Sab05]
- Symmetry Learning Scheme (SLS) [BNOS10]
- Symmetry Propagation (SP) [DBdC^{+}12]
- Symmetry Explanation Learning (SEL) [DBB17]



State-of-the-art dynamic symmetry breaking

State-of-the-art:

- Symmchaff [Sab05]
- Symmetry Learning Scheme (SLS) [BNOS10]
- Symmetry Propagation (SP) [DBdC$^+$12]
- Symmetry Explanation Learning (SEL) [DBB17]



Observations:

Solve some instances very quicly

Cannot handle some instances solved by static approach

Second contribution

Composing Symmetry Propagation and Effective Symmetry Breaking for SAT Solving

NFM'19 [MBK19]

Composing ESBP and SP

Compose the symmetry propagation and the ESBP

prune the decision tree while accelerating its traversal

Composing ESBP and SP

Compose the symmetry propagation and the ESBP

prune the decision tree while accelerating its traversal

Problems:

- ESBP breaks symmetries (incrementally)
- SP considers the manipulated symmetries valid all time

Composing ESBP and SP

Compose the symmetry propagation and the ESBP

prune the decision tree while accelerating its traversal

Problems:

- ESBP breaks symmetries (incrementally)
- SP considers the manipulated symmetries valid all time

In a hybrid approach, SP must be able to identify
valid symmetries

Is valid symmetry?

Our proposal: Local symmetries

Let φ be a formula. We define $L_{\omega,\varphi}$, the set of *local symmetries* for a clause ω, and with respect to a formula φ, as follows:

$$L_{\omega,\varphi} = \{\sigma \in \mathfrak{S} \mid \varphi \vdash \sigma.\omega\}$$

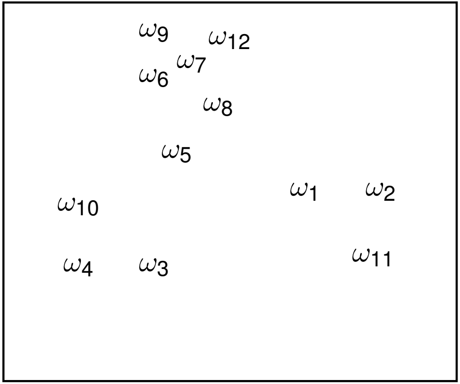
We can state that:

$$\bigcap_{\omega \in \varphi} L_{\omega,\varphi} \subseteq G.$$

Guarantee that symmetrical clauses are logical consequences of the formula

Local symmetry

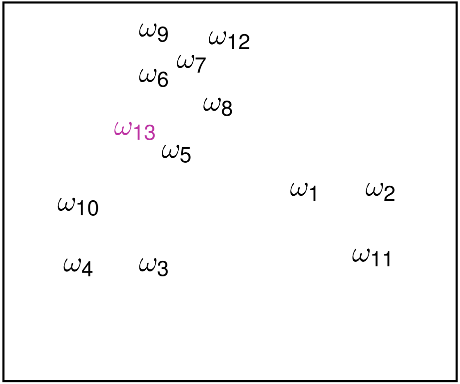
- □ formula
- ω clause
- ω learnt clause



Local symmetries:
$\omega \leftarrow \{g_1, g_2, g_3\}$

Local symmetry

- $\square$ formula
- ω clause
- ω learnt clause
- ω ESBP

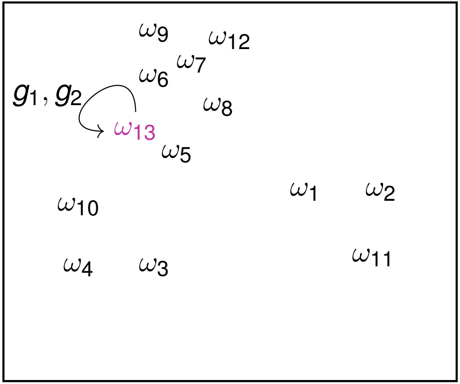


Local symmetries:

$\omega \leftarrow \{g_1, g_2, g_3\}$

Local symmetry

- ◻ formula
- ω clause
- ω learnt clause
- ω ESBP



Local symmetries:

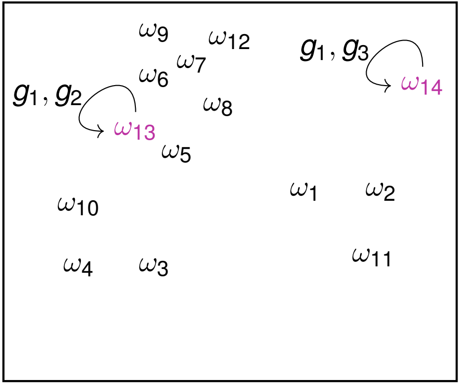
$\omega \leftarrow \{g_1, g_2, g_3\}$

$\omega_{13} \leftarrow \{g_1, g_2\}$

- Compute valid local symmetries
- On the fly
- At minimal cost

Local symmetry

- $\square$ formula
- ω clause
- ω learnt clause
- ω ESBP



Local symmetries:

$\omega \leftarrow \{g_1, g_2, g_3\}$

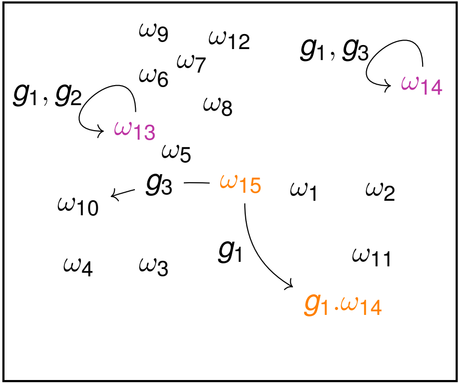
$\omega_{13} \leftarrow \{g_1, g_2\}$

$\omega_{14} \leftarrow \{g_1, g_3\}$

- Compute valid local symmetries
- On the fly
- At minimal cost

Local symmetry

- $\square$ formula
- ω clause
- ω learnt clause
- ω ESBP



Local symmetries:

$\omega \leftarrow \{g_1, g_2, g_3\}$
$\omega_{13} \leftarrow \{g_1, g_2\}$
$\omega_{14} \leftarrow \{g_1, g_3\}$
$\omega_{15} \leftarrow \{g_1, g_3\}$

- Compute valid local symmetries
- On the fly
- At minimal cost

Inductive construction

Experimental results

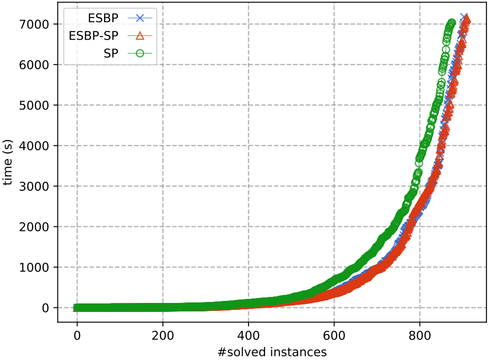
Benchmark:

- from SAT contests 2012 – 2018
- filter: `bliss` finds symmetries in 1000 seconds
- 1400 symmetric instances (out of 4000)

Setup:

- three tools
  - MiniSat SP (Minisat with Symmetry Propagation)
  - MiniSat ESBP (Minisat with cosy)
  - **Minisat ESBP-SP** (our approach)
- 7200 seconds timeout

Experimental results



Solver	PAR-2	SAT	UNSAT
SP	1674h00	406	470
ESBP	1578h30	416	488
ESBP-SP	**1570h15**	**420**	**491**

Discussion of the results

SP and ESBP have separated symmetry managers $\rightarrow$ costly

Discussion of the results

SP and ESBP have separated symmetry managers $\rightarrow$ costly

Combine ESBP with Symmetry Explanation Learning (SEL)

- SEL have less requirements than SP
- We believe that this will improves the performance

Conclusion & Perspectives

Conclusion

- A new dynamic symmetry breaking approach
  - Generation of SBP on the fly
  - Package as a library cosy usable with any CDCL solver
- A new hybrid approach (ESBP-SP)
  - Take advantage of static and dynamic approaches

Conclusion & Perspectives

Conclusion

- A new dynamic symmetry breaking approach
  - Generation of SBP on the fly
  - Package as a library cosy usable with any CDCL solver
- A new hybrid approach (ESBP-SP)
  - Take advantage of static and dynamic approaches

Conclusion & Perspectives

Conclusion

- A new dynamic symmetry breaking approach
  - Generation of SBP on the fly
  - Package as a library cosy usable with any CDCL solver
- A new hybrid approach (ESBP-SP)
  - Take advantage of static and dynamic approaches

Perspectives

- Exploitation of partial symmetries
- Symmetries and parallel SAT solver

Conclusion & Perspectives

Conclusion

- A new dynamic symmetry breaking approach
  - Generation of SBP on the fly
  - Package as a library cosy usable with any CDCL solver
- A new hybrid approach (ESBP-SP)
  - Take advantage of static and dynamic approaches

Perspectives

- Exploitation of partial symmetries
- Symmetries and parallel SAT solver

Thanks !

Fadi A. Aloul, Karem A. Sakallah, and Igor L. Markov.
Efficient symmetry breaking for boolean satisfiability.
IEEE Trans. Computers, 55(5):549–558, 2006.

Belaid Benhamou, Tarek Nabhani, Richard Ostrowski, and Mohamed Reda Saidi.
Enhancing clause learning by symmetry in sat solvers.
In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 329–335. IEEE, 2010.

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.

Jo Devriendt, Bart Bogaerts, and Maurice Bruynooghe.
Symmetric explanation learning: Effective dynamic symmetry handling for sat.
In *International Conference on Theory and Applications of Satisfiability Testing*, pages 83–100. Springer, 2017.

Jo Devriendt, Bart Bogaerts, Maurice Bruynooghe, and Marc Denecker.
Improved static symmetry breaking for sat.
In *International Conference on Theory and Applications of Satisfiability Testing*, pages 104–122. Springer, 2016.

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, 2012.

Martin Davis, George Logemann, and Donald Loveland.
A machine program for theorem-proving.
Commun. ACM, 5(7):394–397, July 1962.

Hakan Metin, Souheib Baarir, Maximilien Colange, and Fabrice Kordon.
Cdclsym: Introducing effective symmetry breaking in sat solving.
In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 99–114. Springer, 2018.

Hakan Metin, Souheib Baarir, and Fabrice Kordon.
Composing symmetry propagation and effective symmetry breaking for sat solving.
In *NASA Formal Methods Symposium*, pages 316–332. Springer, 2019.

Joao P Marques-Silva and Karem A Sakallah.
Grasp: A search algorithm for propositional satisfiability.
IEEE Transactions on Computers, 48(5):506–521, 1999.

Ashish Sabharwal.
Symchaff: A structure-aware satisfiability solver.
In *AAAI*, volume 5, pages 467–474, 2005.

Weakly active symmetries

Logical consequence

When ω is satisfied in all satisfying assignments of φ, we say that ω is a logical consequence of φ, and we denote this by $\varphi \vdash \omega$.

Weakly active symmetries

Logical consequence

When ω is satisfied in all satisfying assignments of φ, we say that ω is a logical consequence of φ, and we denote this by $\varphi \vdash \omega$.

Weakly active symmetries

Let a subset $\delta \subseteq \alpha$, a symmetry σ of φ such that $\varphi \cup \delta \vdash \varphi \cup \alpha \wedge \sigma.\delta \subseteq \alpha$ then σ is weakly active symmetry.

Weakly active symmetries

Logical consequence

When ω is satisfied in all satisfying assignments of φ, we say that ω is a logical consequence of φ, and we denote this by $\varphi \vdash \omega$.

Weakly active symmetries

Let a subset $\delta \subseteq \alpha$, a symmetry σ of φ such that $\varphi \cup \delta \vdash \varphi \cup \alpha \wedge \sigma.\delta \subseteq \alpha$ then σ is weakly active symmetry.

Symmetry propagation

Let σ a weakly active symmetry, then

$$\varphi \cup \alpha \vdash \{l\} \Leftrightarrow \varphi \cup \alpha \vdash \sigma.\{l\}$$

Local symmetries

Logical consequence

When ω is satisfied in all satisfying assignments of φ, we say that ω is a logical consequence of φ, and we denote this by $\varphi \vdash \omega$.

Local Symmetries

Let φ be a formula. We define $L_{\omega,\varphi}$, the set of *local symmetries* for a clause ω, and with respect to a formula φ, as follows:

$$L_{\omega,\varphi} = \{\sigma \in \mathfrak{S} \mid \varphi \vdash \sigma.\omega\}$$

Local symmetries

Logical consequence

When ω is satisfied in all satisfying assignments of φ, we say that ω is a logical consequence of φ, and we denote this by $\varphi \vdash \omega$.

Local Symmetries

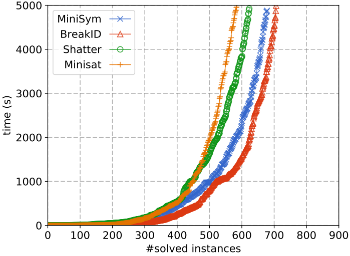
Let φ be a formula. We define $L_{\omega,\varphi}$, the set of *local symmetries* for a clause ω, and with respect to a formula φ, as follows:

$$L_{\omega,\varphi} = \{\sigma \in \mathfrak{S} \mid \varphi \vdash \sigma.\omega\}$$

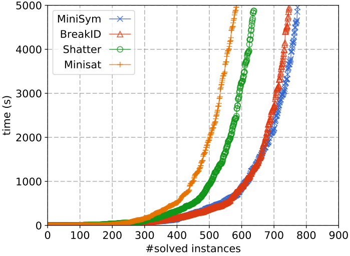
We can state that:

$$\bigcap_{\omega \in \varphi} L_{\omega,\varphi} \subseteq G.$$

Experimental results



(a) with `saucy3`



(b) with `bliss`

Computing local symmetries

Formula can be decomposed as : $\varphi = \varphi_o \cup \varphi_e \cup \varphi_d$ where

- φ_o is the set of the original clauses
- φ_e is the set of ESBPs
- φ_d is the set of deduced clauses.

Local symmetries

- $\omega \in \varphi_o, L_{\omega,\varphi} \supseteq G$
- $\omega \in \varphi_e, L_{\omega,\varphi} \supseteq Stab(\omega) = \{\sigma \in G \mid \omega = \sigma.\omega\}$
- $\omega \in \varphi_d, L_{\omega,\varphi} \supseteq (\bigcap_{\omega' \in \varphi_1} L_{\omega',\varphi}) \cup Stab(\omega)$

  where φ_1 is the set of clauses that derives ω.

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \texttt{F} < \texttt{T}$$

$$g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$$

$$g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP
O_1	F	T	–	–	–	$\cdots$	✓	

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$$

$$g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP
O_1	F	T	–	–	–	$\cdots$	✓	
	T	F	–	–	–	$\cdots$	✗	$\rightarrow \neg x_1 \vee x_2$

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$

$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP
O_1	F	T	–	–	–	$\cdots$	✓	
	T	F	–	–	–	$\cdots$	✗	$\rightarrow \neg x_1 \vee x_2$
O_2	F	F	–	F	T	$\cdots$	✓	

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$$

$$g = (x_1\ \ x_2)(x_4\ \ x_5)(x_7\ \ x_8)$$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP
O_1	F	T	–	–	–	$\cdots$	✓	
	T	F	–	–	–	$\cdots$	✗	$\rightarrow \neg x_1 \vee x_2$
O_2	F	F	–	F	T	$\cdots$	✓	
	F	F	–	T	F	$\cdots$	✗	$\rightarrow x_1 \vee x_2 \vee \neg x_4 \vee x_5$

Generates symmetry breaking predicates (SBP)

- Define lexicographic order
  - Define total order on variables
  - Define minimal value
- Forbid non minimal assignment for each orbit

Example:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8; \mathtt{F} < \mathtt{T}$$

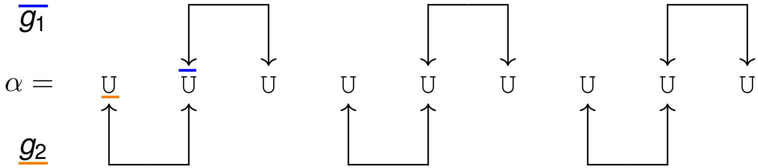
$$g = (x_1 \;\; x_2)(x_4 \;\; x_5)(x_7 \;\; x_8)$$

	x_1	x_2	x_3	x_4	x_5	$\cdots$	lex-leader	SBP
O_1	F	T	–	–	–	$\cdots$	✓	
	T	F	–	–	–	$\cdots$	✗	$\rightarrow \neg x_1 \vee x_2$
O_2	F	F	–	F	T	$\cdots$	✓	
	F	F	–	T	F	$\cdots$	✗	$\rightarrow x_1 \vee x_2 \vee \neg x_4 \vee x_5$
				$\cdots$				

Example

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

$$\mathtt{F} < \mathtt{T} \quad x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \leq x_9$$



Example

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

$\mathtt{F} < \mathtt{T} \quad x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \leq x_9$

$\overline{g_1}$

$\alpha = \quad \underline{\mathtt{U}} \quad \overline{\mathtt{F}} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U}$

$\underline{g_2}$

Example

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

$$\mathtt{F} < \mathtt{T} \quad x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \leq x_9$$

$\overline{g_1}$

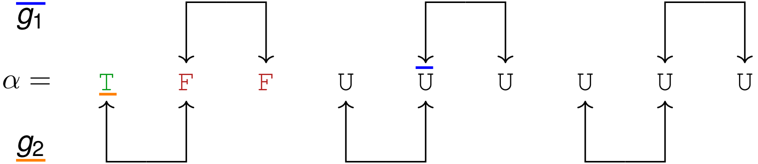
$$\alpha = \quad \underline{\mathtt{U}} \quad \mathtt{F} \quad \mathtt{F} \quad \mathtt{U} \quad \overline{\mathtt{U}} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U}$$

$\underline{g_2}$

Example

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

$$\mathtt{F} < \mathtt{T} \quad x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le x_6 \le x_7 \le x_8 \le x_9$$



Example

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

$$\mathtt{F} < \mathtt{T} \quad x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le x_6 \le x_7 \le x_8 \le x_9$$

$\overline{g_1}$

$$\alpha = \quad \underline{\mathtt{T}} \quad \mathtt{F} \quad \mathtt{F} \quad \mathtt{U} \quad \overline{\mathtt{U}} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U} \quad \mathtt{U}$$

$\underline{g_2}$

g_2 generates ESBP $\omega = \{\neg x_1, x_2\}$

Example

1. `reducer`: $\alpha(g.x) \leq \alpha(x)$
2. `inactive`: $\alpha(x) \leq \alpha(g.x)$
3. `active`: $\alpha(g.x)$ or $\alpha(x)$ is unassigned

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \;\; ; \;\; \mathtt{F} < \mathtt{T}$$

$$\begin{array}{llllllll|ll} g_1 = & (x_2 & x_3) & (x_5 & x_6) & (x_8 & x_9) & & x = x_2 & g.x = x_3 \\ & \uparrow & & & & & & & & \mathtt{active} \end{array}$$

$$\begin{array}{llllllll|ll} g_2 = & (x_1 & x_2) & (x_4 & x_5) & (x_7 & x_8) & & x = x_1 & g.x = x_2 \\ & \uparrow & & & & & & & & \mathtt{active} \end{array}$$

$$\ldots$$

$$\alpha = \{ \qquad\qquad\qquad \}$$

Example

1. `reducer`: $\alpha(g.x) \leq \alpha(x)$
2. `inactive`: $\alpha(x) \leq \alpha(g.x)$
3. `active`: $\alpha(g.x)$ or $\alpha(x)$ is unassigned

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \;\; ; \;\; \mathtt{F} < \mathtt{T}$$

$$g_1 = \quad \underset{\uparrow}{(x_2} \quad x_3) \quad (x_5 \quad x_6) \quad (x_8 \quad x_9) \quad \Big| \quad x = x_2 \quad \underset{\mathtt{active}}{g.x = x_3}$$

$$g_2 = \quad \underset{\uparrow}{(x_1} \quad x_2) \quad (x_4 \quad x_5) \quad (x_7 \quad x_8) \quad \Big| \quad x = x_1 \quad \underset{\mathtt{active}}{g.x = x_2}$$

$$\ldots$$

$$\alpha = \{\neg x_2 \qquad\qquad \}$$

Example

1. `reducer`: $\alpha(g.x) \leq \alpha(x)$
2. `inactive`: $\alpha(x) \leq \alpha(g.x)$
3. `active`: $\alpha(g.x)$ or $\alpha(x)$ is unassigned

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \ ; \ \mathtt{F} < \mathtt{T}$$

$$g_1 = \quad (x_2 \quad x_3) \quad (\underset{\uparrow}{x_5} \quad x_6) \quad (x_8 \quad x_9) \;\Big|\; x = x_5 \quad g.x = x_6 \ \ \mathtt{active}$$

$$g_2 = \quad (\underset{\uparrow}{x_1} \quad x_2) \quad (x_4 \quad x_5) \quad (x_7 \quad x_8) \;\Big|\; x = x_1 \quad g.x = x_2 \ \ \mathtt{reducer}$$

$$\ldots$$

$$\alpha = \{\neg x_2, \neg x_3, x_1\}$$

Example

1. `reducer`: $\alpha(g.x) \leq \alpha(x)$
2. `inactive`: $\alpha(x) \leq \alpha(g.x)$
3. `active`: $\alpha(g.x)$ or $\alpha(x)$ is unassigned

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \;\; ; \;\; \mathtt{F} < \mathtt{T}$$

$$\begin{array}{llllllll|ll} g_1 = & (x_2 & x_3) & (x_5 & x_6) & (x_8 & x_9) & & x = x_5 & g.x = x_6 \\ & & & \uparrow & & & & & & \mathtt{active} \end{array}$$

$$\begin{array}{llllllll|ll} g_2 = & (x_1 & x_2) & (x_4 & x_5) & (x_7 & x_8) & & x = x_1 & g.x = x_2 \\ & \uparrow & & & & & & & & \mathtt{reducer} \end{array}$$

$$\ldots$$

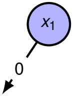
$$\alpha = \{\neg x_2, \neg x_3, x_1\}$$

g_2 generates $\omega = \{\neg x_1, x_2\}$

CDCL in action TODO

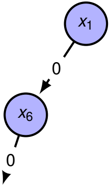
$$
\begin{aligned}
\omega_1 &= \{x_1, x_2, x_3\} \\
\omega_2 &= \{x_4, x_5, x_6\} \\
\omega_3 &= \{\neg x_1, \neg x_5\} \\
\omega_4 &= \{\neg x_2, \neg x_4\} \\
\omega_5 &= \{\neg x_3, \neg x_4\} \\
\omega_6 &= \{\neg x_3, \neg x_6\}
\end{aligned}
$$

CDCL in action TODO



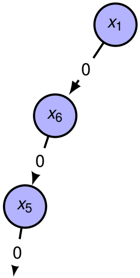
$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



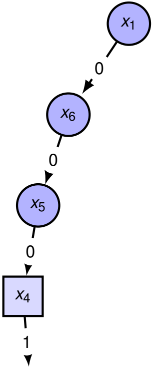
$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



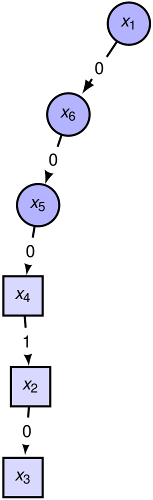
$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



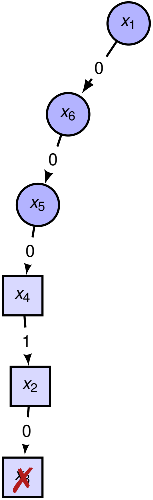
$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



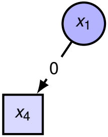
$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$\omega_1 = \{x_1, x_2, x_3\}$$
$$\omega_2 = \{x_4, x_5, x_6\}$$
$$\omega_3 = \{\neg x_1, \neg x_5\}$$
$$\omega_4 = \{\neg x_2, \neg x_4\}$$
$$\omega_5 = \{\neg x_3, \neg x_4\}$$
$$\omega_6 = \{\neg x_3, \neg x_6\}$$

CDCL in action TODO



$$
\begin{aligned}
\omega_1 &= \{x_1, x_2, x_3\} \\
\omega_2 &= \{x_4, x_5, x_6\} \\
\omega_3 &= \{\neg x_1, \neg x_5\} \\
\omega_4 &= \{\neg x_2, \neg x_4\} \\
\omega_5 &= \{\neg x_3, \neg x_4\} \\
\omega_6 &= \{\neg x_3, \neg x_6\}
\end{aligned}
$$

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